Implementation of Filters for Image Pre-processing for Leaf Analyses in Plantations[*]

Jacqueline Gomes Mertes, Norian Marranghello, and Aledir Silveira Pereira

Instituto de Biociências, Letras e Ciências Exatas,
Universidade Estadual Paulista,
São José do Rio Preto, SP, Brazil
jackxcomp@gmail.com, {norian,aledir}@ibilce.unesp.br

Abstract. In this work an image pre-processing module has been developed to extract quantitative information from plantation images with various degrees of infestation. Four filters comprise this module: the first one acts on smoothness of the image, the second one removes image background enhancing plants leaves, the third filter removes isolated dots not removed by the previous filter, and the fourth one is used to highlight leaves' edges. At first the filters were tested with MATLAB, for a quick visual feedback of the filters' behavior. Then the filters were implemented in the C programming language. At last, the module as been coded in VHDL for the implementation on a Stratix II family FPGA. Tests were run and the results are shown in this paper.

Keywords: Precision agriculture, hardware description language application, digital image processing, reconfigurable architectures.

1 Introduction

Currently, digital image processing techniques are used to decide a variety of problems. Such techniques can improve image visualization for analysis by humans, as, for example, the biomedical interpretation of rays X images. It can also help recognizing irregular products in a production mat in an automated industry [1].

One of the newest uses of image processing techniques came into being with the need for real time processing. One such use is in the agricultural area, aiming at extracting quantitative information from plantations' images with diverse degrees of infestations by various biological agents. In this case it is necessary a specific image treatment to help detecting, and determining the degree of infestation.

In agriculture, the control of weeds is of real importance when the cost of herbicides is taken into consideration. A careful study of the plantation the amount of herbicides to be applied can be considerably reduced. This implies a reduction of the

[*] This work has been partially supported by the Brazilian National Research Council (CNPq) through a project within Brazil-IP program, and by Sao Paulo State Research Foundation (FAPESP) through travel grant number 2008/01645-7.

M. Bubak et al. (Eds.): ICCS 2008, Part II, LNCS 5102, pp. 153 – 162, 2008.

final product as well as prevents unneeded herbicides from being introduced into the environment.

Precision agriculture appeared in the late eighties to reduce fertilizer application according to previous chemical analysis of soil composition [2]. With time new equipments and techniques were developed to improve these studies, and maximize productivity.

Thus, one of the main objectives of the precision agriculture is the adequate handling of the desired culture to maximize productivity, and to minimize the impact on the environment.

PERNOMIAN reports a work in which weeds are identified in real time using artificial neural nets [3]. YANG et al. present another application of artificial neural nets to weed recognition and classification [4]. DIAS developed an application for rust identification in sugarcane plantations also using artificial neural nets [5].

In this paper we present a technique based on image filtering algorithms for weed detection in soybean plantations. A hardware module for leaf image pre-processing is developed based on such algorithms. After described in VHDL the module is simulated and tested. Then it is implemented onto an FPGA. In the following sections each filter is functionally presented then the development of the hardware module is described, and some test results are discussed.

2 Development of the Filters

In order to develop the soybean leaves' images pre-processing module four algorithms were elaborated. The module should identify the leaves from the scene, and highlight their edges so that in the future an artificial neural network (ANN) may use this information to find out possible weeds.

Image dimension was chosen towards using the least possible number of pixels still enabling to properly identify the eventual presence of weeds. When submitted to the ANN each pixel will be treated by an input node of the net. The smaller is the number of pixels, the smaller the total input node count will be. Thus, it is important to keep the number of pixels small so that the implementation of the system comprising the pre-processing module plus the ANN onto an FPGA is viable. Therefore, the images used were restricted to 40x60 pixels, an amount of pixels that is small enough to be implemented on an FPGA, while preserves a good amount of information from the original image.

The pre-processing module was developed with the following functionalities:

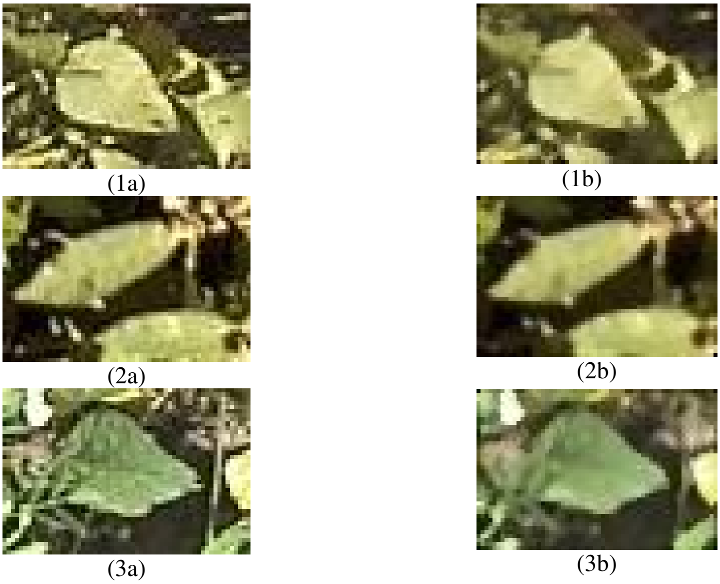
- noise smoothing in the original image through a median filter;
- putting leaves' images apart from the background through image segmentation;
- removing spots not detected by previous processing through a custom designed filter; and
- edge enhancing as well as binarization of the resulting image through a Roberts filter.

At first these filters were implemented with MATLAB as this software has an easy to use language that is also fast to compile, providing prompt results. Then, the algorithms were implemented both in C and in VHDL. We had to face the fact that the VHDL language can only process uni-dimensional matrix declarations. Thus, the three dimensional RGB input matrix had to be broken down to three uni-dimensional vectors with 2400 positions each. Therefore, both C and VHDL implementations of the algorithms dealt with the original image as three vectors, one for each fundamental color R, G, and B.

2.1 Filter for Noise Smoothing

The median filter is used to reduce image noise. In this work it is implemented using a 3x3 pixels' neighborhood. Thus, a 3x3 mask scans the image from an initially defined pixel. The pixels within the 3x3 neighborhood of such an initial pixel are ordered by value. The, the pixel with the same number of values bellow and above it is identified as the median pixel. The mask is then centered onto this new pixel.

Fig. 1 displays three images to show the diversity of leaves that can be found in soybean crops. In all instances the picture to the left (a) is the original image before being submitted to the filter, and the picture to the right (b) is the corresponding image after being processed by the median filter. In Fig. 1 instances 1 and 3 are soybean leaves, and instance 2 is weed. From these example images one can fell the difference among the leaves existing in a soybean plantation.



(1a) (1b)

(2a) (2b)

(3a) (3b)

Fig. 1. Original images of leaves to the left; to the right leaves resulting from median filtering

2.2 Filter for Background Removal

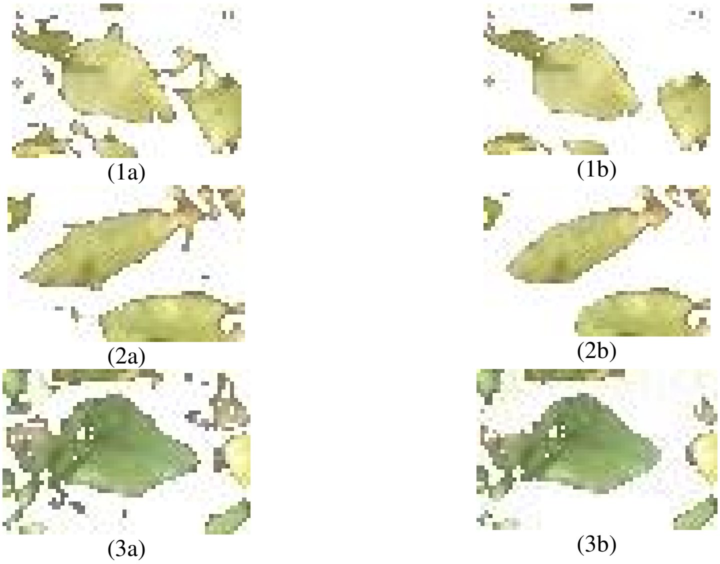
Image segmentation provides for background removal. This means that this filter removes all background information such as soil and other particles laid on the ground hopefully leaving only the leaves in the scene. In order to do that, intervals of color values from the background image are determined. The image is then scanned a pixel at a time to check if the pixel's value is within one of the background intervals. When there is a match the value of the pixel is set to white, it is left unchanged otherwise.

Fig. 2(a) displays the resulting images after processing by the background removal filter. The input images for this filtering step are the corresponding numbered images to the right side of Fig. 1.

2.3 Filter for Noise Removal

This filter aims at removing isolated spots from the image which were not detected by the previous filters. To accomplish that it uses a 7x7-pixel mask. Using this mask to scan the image the amount of white pixels sitting within the mask area is counted. We consider the existence of 25 or more white pixels under the mask area as an indicative of a background area. Thus, when this count is greater then or equal to 25 the pixel corresponding to the center of the mask is regarded as noise, and set to white too.

Fig. 2(b) displays the results from submitting the corresponding images from Fig. 2(a) to the noise removal filter.


(1a) (1b)

(2a) (2b)

(3a) (3b)

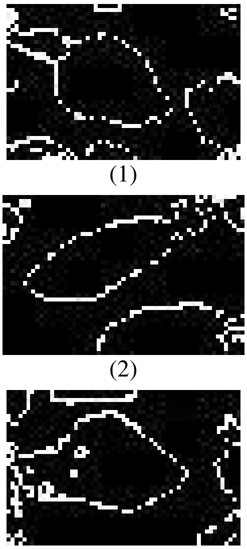
Fig. 2. Images resulting from background removal filtering to the left; to the right images resulting from noise removal filtering

2.4 Filter for Edge Enhancing

The goal of this filter is to enhance the edges of the leaves as well as to turn these edges into binary form to make the image suitable for ANN processing. Some options were considered for edge enhancing, being the one using Roberts' filter that presented the best performance, and the lowest computational cost. Other options considered were those using Sobel, Prewitt, and Nevatia-Babu filters.

In figure 3, one can observe the images resulting from the application of the edge enhancing filter. The images resulting from the noise removal filtering have their edges enhanced. The black-and-white format is the result of the binarization process. From figure 4 the differences between soybean leaves and weeds can be noted. Weed leaves are more elongated while soybean leaves present a more rounded shape.

It is also interesting to note that figure 3 is also the final result of the image pre-processing module. We intend to submit this sort of images to an ANN to tell weeds and soybean leaves apart so that we can determine the amount of herbicides to apply to the crop.



(1)

(2)

(3)

Fig. 3. Images resulting from edge enhancing filter

2.5 Application of the Module of Pre-processing in Real Environment

The module presented above was applied to images of soy plantations to prove its effectiveness. Such images were collected with the aid of a digital camera, at about 1,5 m hieght, perpendicularly to the ground. These images have dimension of 360x240 pixels. In Fig. 4 the result of the pre-processing carried out with one of the collected images is presented.

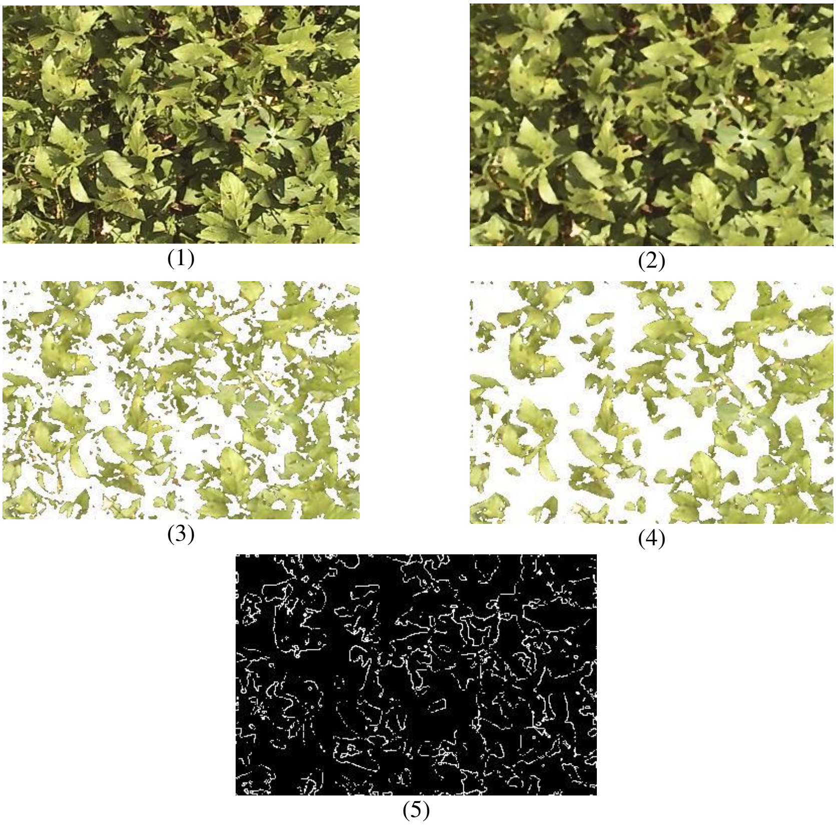

(1)
(2)
(3)
(4)
(5)

Fig. 4. Result of pre-processing: (1) original image; (2) image resulting from median filtering; (3) image resulting from background removal filtering; (4) resulting from noise removal filtering and (5) image resulting from edge enhancing filter

3 Composition of the Filters in VHDL

The algorithms presented in the previous section were implemented in VHDL. The integration of these filters in a module to produce a resulting image that can be directly read by an ANN for weed identification requires some adaptation as well as the creation of auxiliary structures for testing on an FPGA.

Fig. 5 illustrates how the module with the filters was organized for implementation, and how their internal logic blocks communicate to each other. In the following we describe each such structure comprising the image pre-processing module.

From Fig. 5 one can observe the input data that are composed by the signals clock, reset, enable and by the set of pixels of the image. This set of pixels, as previously explained, is comprised by three vectors, namely R, G, and B. These signals will be reach the FPGA though input ports automatically determined by Quartus II software, during programming procedure.

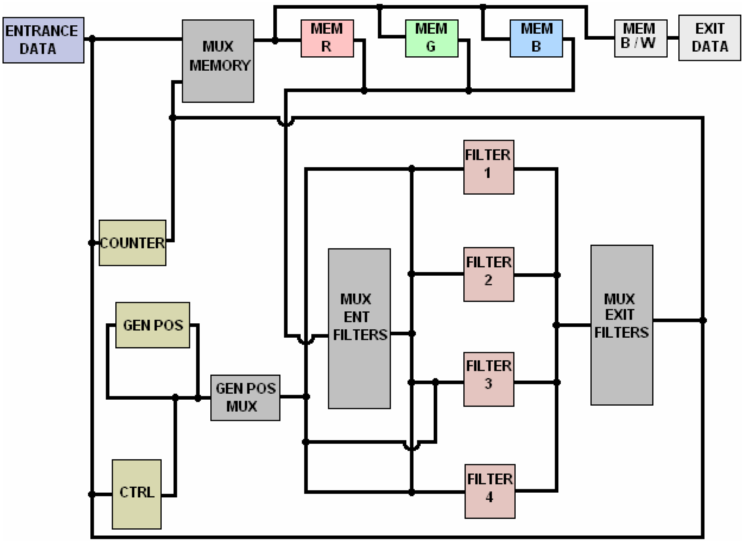
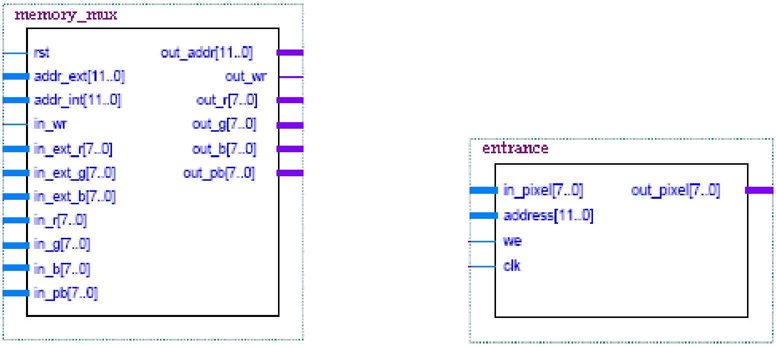


Fig. 5. Block diagram for the implementation of the filters in VHDL

The MUX MEMORY structure is a multiplexer for the signals that must be sent to the internal memories. According to the control signals received the multiplexer switches between the FPGA input ports (corresponding to external data input), and one of the filters outputs (corresponding to internal data inputs). This multiplexer can be seen in the Fig. 6(a).



(a) (b)

Fig. 6. Data memory multiplexer (a); and connections of memory blocks (b)

Blocks MEM R, MEM G, MEM B represent the memories for the storage of information regarding each image color, and MEM P/B represents the memory to store the image resulting from the image enhancing filter. These memories have the same structure, being their external connections presented in Fig. 6(b).

The CONT structure represents an internal counter that is started with the reset signal to sum up the memory positions used by the input data. The counter is shown in Fig. 7(a).

The GEN POS structure represents the pixel position generator used by the median, noise removal, and edge enhancement filters. This is the logic that properly initializes the required mask, as well as moves it by incrementing its position relatively to the corresponding interval. The position generator can be observed in Fig. 7(b).

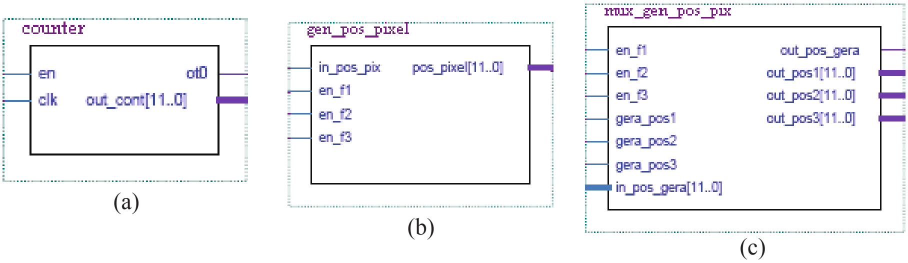


Fig. 7. Counter of memory positions (a); and pixel position generator (b); position generator multiplexer (c)

The MUX GEN POS structure represents the signal multiplexer of the position generator. Thus, it receives signals from the filters, and sends to the position generator the information on the position that has to be incremented. The position generator returns the updated position to the multiplexer, which in turn transmits this information to the proper filter. This multiplexer can be observed in Fig. 7(c).

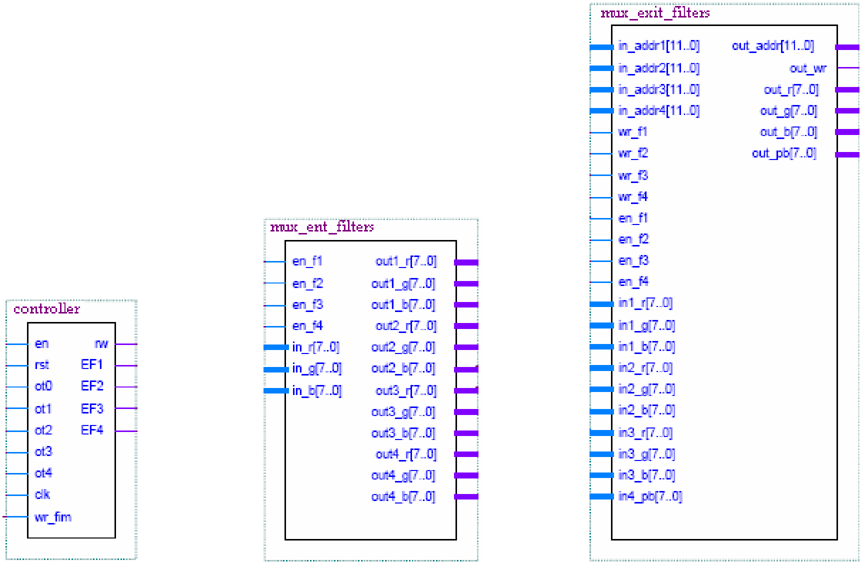
The CTRL structure represents the controller for filter enabling. It enables the first filter, the median filter, when it receives a signal from the internal counter indicating that all memories are filled up. For the other filters to be enabled this controller waits until it receives a task completion signal from the corresponding previous filter. This controller is shown in Fig. 8(a).

The MUX IN FILTERS structure represents the multiplexer of the signals to be sent to the filters. According to the enabled filter it determines the data to be sent. This multiplexer is shown in Fig. 8(b).

The MUX OUT FILTERS structure represents the multiplexer of the signals that are received from the filters during or after processing of the image vectors. This multiplexer is shown in Fig. 8(c).

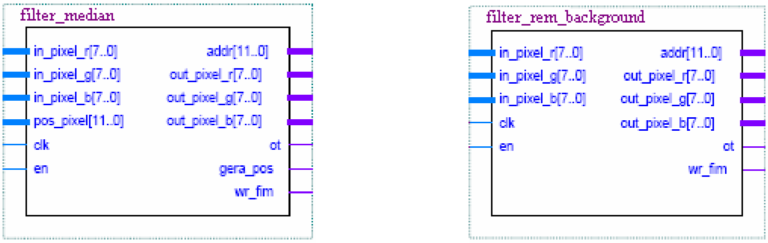
Furthermore, Fig. 5 shows four structures named FILTER 1, FILTER 2, FILTER 3, and FILTER 4, corresponding to the median, background removal, noise removal, and edge enhancement filters. Their signals are presented in Fig. 9(a), 9(b), 10(a), and 10(b), respectively.

Based on the given description of the interconnection of the sub-system in figure 6 the image pre-processing can be described as follows: upon concluding data input the counter sends the signal ot0 to the controller that enables the median filter; finishing its processing this filter sends the signal ot1 to the controller that enables the background removal filter; this filter processes the required data and then sends the signal



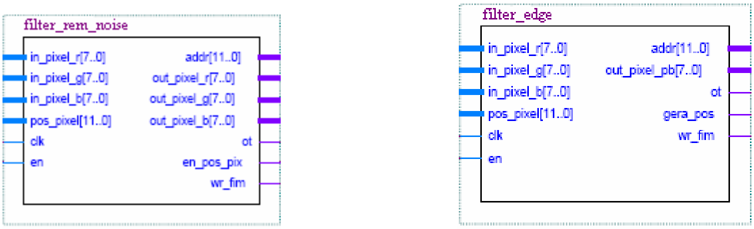
(a) (b) (c)

Fig. 8. Signal controller (a); multiplexers of input (b), and output (c) signals for the filters



(a) (b)

Fig. 9. Median filter (a); filter for background removal (b)



(a) (b)

Fig. 10. Filter for noise removal (a); filter for edge enhancement (b)

ot2 to the controller, which enables the noise removal filter; at last, when the noise removal filter is done with its processing it sends the signal ot3 to the controller that finally enables the edge enhancement filter. Upon completion this filter sends the signal ot4 to the controller to conclude the processing.

Before sending the end-of-processing signal (signoff), each filter stores the resulting vectors in the memories represented for structures MEM R, MEM G and MEM B. Except for the edge enhancement filter, which stores the resulting vector in the memory represented by the structure MEM P/B.

Before sending the end-of-processing signal (signoff), each filter stores the resulting vectors in the memories represented for structures MEM R, MEM G and MEM B. Except for the edge enhancement filter, which stores the resulting vector in the memory represented by the structure MEM P/B.

4 Conclusions

In this article filters for image pre-processing of soybean plantations were presented. Identification of such images through their edges was possible mainly due to the smart performance of image segmentation filters, and the easy edge detection made possible by the edge enhancement filter.

From the results presented we intend to move on by designing and training an artificial neural net capable of identifying weeds in images like these. After effective weed detection it is possible to determine the correct amount of herbicides to be applied in the considered crop. Thus, reaching the goals of precision agriculture.

References

1. Gonzalez, R.C., Woods, R.E.: Processamento de Imagens Digitais (1993)
2. Whelan, B.M.: Precision Agriculture – An Introduction to Concepts, Analysis & Interpretation. ACPA, Sydney, Australia, p. 154 (2005)
3. Pernomian, V.A.: Identificação de plantas invasoras em tempo real. ICMC, USP. São Carlos (2002)
4. Yang, C.C., et al.: Application of artificial neural networks in image recognition and classification of crops and weeds. Canadian Agricultural Engineering 42(3) (July 2000)
5. Dias, D.N.: Identificação dos sintomas de ferrugem em áreas cultivadas com cana-de-açúcar. ICMC, USP. São Carlos (2004)