System-on-a-chip (SoC) FPGA-based Hardware Acceleration for Green Crop Segmentation

Shan Li, Zhibin Zhang and Yuan Zheng
School of computer science, Inner Mongolian University, Hohhot, 010021, China
Email: 534896988@qq.com
Email: cszhbin@imu.edu.cn
Email: 454388711@qq.com

Abstract. With the development of artificial intelligence technology, agricultural robot plays a significantly important role for agricultural intelligence. Crop segmentation is a critical and fundamental pre-processing step for agricultural robot navigation and crop growth monitoring. Although there are several crop segmentation methods, they cannot meet the online real-time requirement for agricultural robot. In view of this, a real-time crop segmentation system implemented on a SoC-FPGA is first proposed in this paper, which contains image loading module, segmentation algorithm module and image display module. As a core module, segmentation algorithm module consists of greenness identification, thresholding operation and morphological operation. Its hardware implementation adopted a parallel pipeline architecture to enhance real-time performance. The proposed system is implemented on a Xilinx Zynq-7000 FPGA and the experimental results showed that it can achieve a frame rate of 60 frames per second for the image resolution of 1920×1080 with the average accuracy of 91%.

1. Introduction

As an emerging technology, artificial intelligence has gradually penetrated into various fields including agriculture field. In the field of agriculture, agricultural robot, instead of human labors, plays a significantly important role for agricultural intelligence gradually, and thus has become an active area of research in recent years [1, 2]. Generally, agricultural robots are equipped with cameras to capture the images of crops. From crop images, crop segmentation aims to distinguish green crop from its background (e.g. soils and rocks) which is an essential step for crop row detection, agricultural robot vision navigation and crop growth monitoring. Although there are some researches on image segmentation for general images [3, 4], they are not exactly suitable for green crop segmentation.

According to existing platform implementation, the crop segmentation methods can be divided into two types: CPU-based software implementation and DSP-based hardware implementation. For the first kind of methods, it can be exactly divided into the feature descriptor-based methods [5-10] and the machine learning-based methods [11, 12]. Excess green index (ExG) [5], color index of vegetation extraction (CIVE) [6] and vegetative index (VEG) [7] are the linear combination, the weighted sum and the nonlinear combination of pixel values in red, green and blue channels, respectively. A combination of ExG, CIVE and VEG (COM) was proposed in [8-10] which is a weighted sum of these three indices. Unlike the above feature descriptor-based methods, the machine learning-based methods aim to employ various learning algorithms to distinguish green crop from background. The back propagation neural network with particle swarm optimization is utilized in [11] and the Bayesian decision-making is utilized in [12]. The existing crop segmentation methods with hardware implementation are mainly based on digital signal processing (DSP) [13, 14]. The limitation of these
methods resides in that the execution time cannot meet the real-time requirement for agricultural robot vision navigation and crop growth monitoring.

Recently, FPGA has received great attention due to the performance of high speed and low power consumption, and thus it is broadly utilized as a hardware accelerator [15, 16]. In view of this, this paper firstly presents a real-time crop segmentation system implemented on a Xilinx Zynq-7000 SoC-FPGA, which can achieve a frame rate of 60 frames per second at the image resolution of 1920×1080 with an average accuracy of 91%. The proposed system consists of three modules: image loading module, segmentation algorithm module and image display module. As a core module, the segmentation algorithm module contains greenness identification, thresholding operation and morphological operation. Its implementation adopts a parallel pipeline architecture in order to enhance real-time performance, where line buffers and sliding window are used. The benefit of doing this is to reduce access to DDR memory, increase the bandwidth of data transmission, and further improve the performance of the whole system.

2. Crop Segmentation Algorithm
Generally, green crop segmentation algorithms consist of three steps: greenness identification, thresholding operation and morphological operation, as shown in Figure 1.

![Figure 1. Block diagram of green crop segmentation algorithms](image)

2.1. Greenness Identification
Greenness identification aims to emphasis green crop from non-green background. Here, we choose ExG descriptor [5] for greenness identification due to its good robustness and illumination invariance, which converts an RGB image into a gray image by the following formula:

$$ExG = (2 \times G) - R - B$$  \hspace{1cm} (1)

where $R$, $G$ and $B$ are the pixel values in red, green and blue channels, respectively. Notice that the value of ExG has a range from 0 to 255 via normalization.

2.2. Thresholding Operation
Thresholding operation is to compute a threshold and then distinguish green crop from non-green background by using this threshold. Here, we utilize Otsu's method [17] due to its well-tested performance, where the threshold is calculated by maximizing the inter-class variance between foreground class and background class, namely.

$$T = \arg \max_t \omega_0(t) \sigma_0^2(t) + \omega_1(t) \sigma_1^2(t)$$  \hspace{1cm} (2)

where $t$ denotes the threshold to be optimized ranging from 0 to 255, $\omega_0(t)$ and $\omega_1(t)$ denote the probabilities of these two classes, and $\sigma_0^2(t)$ and $\sigma_1^2(t)$ denote the variances of these two classes.

2.3. Morphological operation
Filtering operation aims to remove noise pixels over the binary image, leading to a cleaner image than the previous one. Here, median filtering is adopted that helps eliminate noise and preserves edges [18], and a sliding window of 5×5 is used. For median filtering, the filtered value at the $i$-th pixel point, denoted by $y_i$, can be expressed by the following formula:

$$y_i = \text{med}(f_{i-k}, \ldots, f_i, \ldots, f_{i+k})$$  \hspace{1cm} (3)

where $f_{i-k}, \ldots, f_i, \ldots, f_{i+k}$ denote the pixel values at pixel points within the sliding window, and $\text{med}()$ denotes the function to calculate the median of these pixel values within the sliding window.
3. System Architecture and Implementation

3.1. System Profile
The proposed crop segmentation system implemented on a Xilinx SoC-FPGA board is shown in Figure 2 (a). The processing system (PS) and the programmable logic (PL) are connected via advanced extensible interface (AXI) bus, and the interfaces of AXI bus include accelerator coherency port (ACP) and general purpose ports (GP0). The video stream data is transferred from PS to PL via video direct memory access (VDMA), and the crop segmentation algorithm is implemented in PL.

Figure 2 (b) depicts the data flow of the crop segmentation system from image loading module, segmentation algorithm module to image display module, where image loading module is to load video data to DDR memory, segmentation algorithm module is to distinguish crops from background and image display module is to output segmentation image to display devices.

3.2. Implementation of Segmentation Algorithm
The implementation of segmentation algorithm module is split into four parts: greenness computation, threshold computation, crop foreground computation and median filtering. Greenness and threshold computation are implemented in parallel shown in yellow frame of Figure 2 (b); crop foreground computation and median filtering are implemented in parallel shown in green of Figure 2 (b).

Greenness and threshold computation is to read the pixel values from DDR memory and calculate the values of $ExG$ at each pixel according to Eq. (1), and implement multiply operation with shift register; at the same time, the threshold used for crop segmentation is iteratively calculated and used multiplexer to select the values of $ExG$. Figure 3 (a) describes the implementations of greenness and threshold computation module in detail.

Crop foreground computation is to compare pixel value with the threshold, if the pixel value is greater than the threshold, it is regarded as crop foreground, otherwise as background; at the same time, median filtering with 5×5 sliding window is implemented and the filtered values are obtained by counting the pixel points representing crop foreground. Figure 3 (b) describes the implementations of
crop foreground computation and median filtering in detail, where a parallel pipeline architecture is adopted. In Figure 5, line buffers are used to cache the pixel values from five rows of the image, and the sliding window of 5×5 is to collect 25 pixel values from line buffers. Once the sliding window is filled up, the median filtering starts to implement rather than after the crop foreground computation of the whole image. The benefits of doing these consist in the improvement of real-time performance.

![Crop segmentation results](image.png)

**Figure 4.** Proposed crop segmentation system and the results.

### 4. Result and Evaluations

The proposed crop segmentation system is implemented on a Xilinx Zynq-7000 FPGA board with Vivado 2017.4, as shown in Figure 4 (a). In order to evaluate the proposed system, accuracy, running time and resource utilization are discussed.

First, we test the accuracy of the proposed crop segmentation system on our self-collecting image sets and the data sets provided by [19] (totally 300 original crop images). In order to verify the accuracy, we compare the crop segmentation images with the ground-truth images, and then calculate the following three evaluation indexes:

\[
\text{precision} = \frac{TP}{FP + TP} \tag{4}
\]

\[
\text{recall} = \frac{TP}{TP + FN} \tag{5}
\]

\[
F - \text{score} = \frac{2 \times TP}{2 \times FP + TP + FN} \tag{6}
\]

where TP is the number of positives, FP is the number of false positives, and FN is the number of false negatives.

Table 1 lists these three evaluation indexes obtained using the proposed crop segmentation system, where various situations including crop types, illumination conditions and image qualities are considered. From Table 1, it can be seen that the proposed system can achieve an average F-score of 91.1% when considering both ideal situations and non-ideal situations.

Figure 7 presents several typical crop segmentation images obtained using the proposed system. Figure 7 (b)-(d) give the segmentation images of greenhouse strawberry crops under three different illumination conditions; Figure 7 (e)-(g) give the segmentation image of outdoor beet crops under different illumination conditions and image qualities; Figure 7 (h) gives the segmentation image of outdoor cabbage crops, where exist obvious shadows and reflections due to the large leaves of the cabbage; Figure 7 (i) gives the segmentation image of outdoor corn crops, where exist obvious background noises including pedestrians, taxi and trees in the distance.

Next, we test the time consumption of the proposed crop segmentation system and its software implementation based on a 3.4 GHz CPU. Here, we compare with the software implementation in Qt 5.6 under single thread and 16 threads, and the range of image resolutions is from 640×480, 1080×720 to 1920×1080. Table 2 lists the averages of running time obtained by the proposed system and its software implementation on a 3.4 GHz computer. As can be seen from Table 2, the proposed hardware implementation is obviously faster than the software implementation; moreover, when the image
resolution is 1920×1080, the proposed hardware implementation is faster at least 9x and 6x than the software implementation under single thread and 16 threads, respectively.

Finally, we evaluate the resource utilization of the proposed crop segmentation system on a Xilinx Zynq-7000 FPGA board. Table 3 lists the resource utilization of the proposed segmentation algorithm module and the whole system when the image resolution is 1920×1080. As we can see, the segmentation algorithm module only occupies 8% of flip-flops (FFs), 42% of look-up tables (LUTs) and 14% of BRAMs, while the whole system occupies 47% of FFs, 66% of LUTs and 24% of BRAMs. This result shows that the whole system does not take up resources too much, and it is possible to integrate other vision tasks into the proposed system.

Table 1. Results obtained using the proposed crop segmentation system in terms of accuracy

| Evaluation indexes | Average for ideal situations | Average for non-ideal situations | Whole average |
|--------------------|------------------------------|---------------------------------|---------------|
| Precision          | 90.5%                        | 80.8%                           | 85.2%         |
| Recall             | 98.5%                        | 97.8%                           | 98.1%         |
| F-score            | 94.3%                        | 88.4%                           | 91.1%         |

Table 2. Comparison of running time of the proposed hardware implementation on FPGA with software implementation on a 3.4 GHz CPU (unit: milliseconds)

| Image resolution | CPU with 1 thread | CPU with 16 threads | The proposed system |
|------------------|-------------------|---------------------|---------------------|
| 640×480          | 61.1              | 21.1                | 2.4                 |
| 1280×720         | 88.5              | 43.9                | 7.1                 |
| 1920×1080        | 158.2             | 98.3                | 16.0                |

Table 3. Resource utilization on Xilinx Zynq-7000 at the image resolution of 1920×1080

| Resources | Available | Segmentation Algorithm module usage | Whole segmentation system usage |
|-----------|-----------|------------------------------------|---------------------------------|
| FF        | 35200     | 2378 (8%)                          | 16497 (47%)                     |
| LUT       | 17600     | 7232 (42%)                         | 11522 (66%)                     |
| BRAM      | 60        | 8 (14%)                            | 14 (24%)                        |

5. Conclusions
This paper firstly proposes a real-time crop segmentation system implemented on a Xilinx SoC-FPGA, which contains image loading module, segmentation algorithm module and image display module. As a core module, the segmentation algorithm module is comprised of greenness identification, thresholding operation and morphological operation. In order to improve real-time performance, its implementation adopts a parallel pipeline architecture, where line buffers and sliding window are used. The experimental results on real crop images show that the proposed crop segmentation system can achieve a frame rate of 60 frames per second for an image resolution of 1920×1080 with an average accuracy of 91%. Such results demonstrate that the proposed system is able to meet online real-time requirement for agricultural robot vision navigation and crop growth monitoring.

6. Acknowledgments
This work is supported by the National Natural Science Foundation of China(Nos.31760345, 61502119).

7. References
[1] Vidović I, Cupec R. and Hocenski Ž 2016 Crop Row Detection by Global Energy Minimization. Pattern Recognition. 55 68-86.
[2] Hossein Mousazadeh 2013 A technical review on navigation systems of agricultural autonomous off-road vehicles. Journal of Terramechanics. 50 211-232.
[3] Perciano T, Ushizima D M, Bethel E W, Mizrahi Y D, Parkinson D and Sethian J A 2016 Reduced-complexity image segmentation under parallel Markov Random Field formulation using graph partitioning. *IEEE Int. Conf. on Image Processing (ICIP)*. Phoenix. p 1259-1263.

[4] Fu X, Chen C, Li J, Wang C and Kuo C J 2017 Image segmentation using contour, surface, and depth cues. *IEEE Int. Conf. on Image Processing (ICIP)*. Beijing. p 81-85.

[5] Woebbecke D M, Meyer G E, Bargen K V and Mortensen D A 1995 Color indices for weed identification under various soil, residue, and lighting conditions. *Transactions of the American Society of Agricultural Engineers*. 38 259-269.

[6] Kataoka T, Kaneko T, Okamoto H, and Hata S 2003 Crop growth estimation system using machine vision. *Proc. IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics (AIM 2003)*. Hong Kong. p 1079-1083.

[7] Hague T, Tillett N D and Wheeler H 2006 Automated crop and weed monitoring in widely spaced cereals. *Precision Agriculture*. 7 21-32.

[8] Guijarro M, Pajares G, Riomoros I, Herrera P J, Burgos-Artizzu X P and Ribeiro A 2011 Automatic segmentation of relevant textures in agricultural images. *Computers and Electronics in Agriculture*. 75 75-83.

[9] Guerrero J M, Pajaresa G, Montalvo M, Romeo J and Guijarro M 2012 Support Vector Machines for crop/weeds identification in maize fields. *Expert Systems with Applications*. 39 11149-11159.

[10] Romeo J, Pajares G, Montalvo M, Guerrero J M, Guijarro M and de la Cruz J M 2013 A new Expert System for greenness identification in agricultural images. *Expert Systems with Applications*. 40 2275-2286.

[11] Bai X, Cao Z, Wang Y, Yu Z, Hu Z, Zhang X and Li C 2014 Vegetation segmentation robust to illumination variations based on clustering and morphology modelling. *Biosystems Engineering*. 125 80-97.

[12] Tang J, Chen X, Miao R and Wang D 2016 Weed detection using image processing under different illumination for site-specific areas spraying. *Computers and Electronics in Agriculture*. 125 103-111.

[13] Mancini A, Dyson J, Frontoni E and Zingaretti P 2017 Soil/crop segmentation from remotely sensed data acquired by Unmanned Aerial System. *Int. Conf. on Unmanned Aircraft Systems*. USA. p 1410-1417.

[14] Yin D, Chen S Pei W, Zhou X and Chen Z 2011 Experiment study of real-time recognition of weeds in cotton field based on DSP. *Int. Conf. on New Technology of Agricultural*. p 258-262.

[15] Patil S R and Gawade N M 2015 Implementation of segmentation of blood vessels in retinal images on FPGA. *Int. Conference on Information Processing (ICIP)*. Pune. p 516-519.

[16] Puglia L, Vigliari M and Raiconi G 2017 Real-Time Low-Power FPGA Architecture for Stereo Vision. *IEEE Transactions on Circuits and Systems II: Express Briefs*. 11 1307-1311.

[17] Garcia-Lamont F, Cervantes J, López A and Rodriguez L 2018 Segmentation of images by color features: A survey. *Neurocomputing*. 292 1-27.

[18] Onyango M, Hester P Y, Stroshine R L and Adeola O 2003 Bone densitometry as an indicator of percentage tibia ash in broiler chicks fed varying dietary calcium and phosphorus levels. *Poultry Science, Poultry Science Association*. 82 1787-1791.

[19] Santillán I G, Montalvo M, Guerrero J M and Pajares G 2017 Automatic detection of curved and straight crop rows from images in maize fields. *Biosystems Engineering*. 156 61-79.