Using Fast Fourier Transform for Weed Detection in Corn Fields

Hossein Nejati
School of Computing
National University of Singapore
Singapore, Singapore
nejati@nus.edu.sg

Zohreh Azimifar
Computer Science and Engineering
Shiraz University
Shiraz, Iran
azimifar@shirazu.ac.ir

Mohsen Zamani
Electrical and Computer Engineering Dept
National University of Singapore
Singapore, Singapore
zamani@nus.edu.sg

***Abstract*— Agriculture plays one of the most important roles in economy and therefore lowering costs and improving quality of agricultural products is highly demanded. Controlling weeds is one of the most important and also expensive labors in agriculture which can be automated using robotic cultivators. These robots should be armed with a digital camera which uses a method to classify between weeds and crops based on captured image and then remove the weed by spraying herbicide accurately on the weed, cutting with blades or damaging with electric shock devices. Any of these methods reduce herbicide usage which also protects environment from side-effects of these chemical substances.**

In this article a method is proposed which utilizes fast Fourier transform and leaf edge density to classify between crop and weed leaves in corn fields in real-time. This method is based on specific shapes of these leaves and leaf vein structures. Testing the method on a sample set of corn field images showed more than 92% accuracy in detecting weed plants. The resulting application is finally compiled to a dynamic linked library (dll) and used in a graphical user interface (GUI) to be used further by a cultivator robot in a real field.

I. Introduction

Agriculture plays one of the most important roles in economy of countries. This fact has led to different approaches toward lowering the costs and improving the quality of products in agricultural industry. One of the most important and also costly labors in this industry is controlling weeds. As weed can be defined as every plant which has grown in an inappropriate place, weed controlling can be extended from farms to lawns, golf fields, and sport fields. Every year a large amount of herbicide is used for removing weeds from agricultural fields which is not only expensive, but also a source of environmental pollution. Hand labor is also costly and time consuming. Therefore, many researches have been done to utilize computer vision and robotics to develop robotic cultivators for instantaneous weed detection and removal.

Methods applied for weed detection are usually have three main phases: 1. Capturing a top-view image and background segmentation using plants distinguishable green color; 2. Noise removal and pre-processing; And at last, 3. Extracting features of each plant for weed/crop classification. This results to herbicide usage reduction to even 100% when mechanical or electrical devices are used for weed removal. Examples of this approach are [1] used in carrot fields and [2], [3] used in tomato fields.

Another work in lawn fields is proposed by Watchareeruetai et al. (2006) using three detection methods of: Edge detection, Bayesian classifier detection, and morphological classification [4]. In the edge detection method, the fact that grass blades cause considerably more edges to appear in a unit square of field image in comparison with weeds is utilized for classification. This is because weeds (in lawn fields) have larger leaves, resulting in less edges after edge detection techniques.

In this paper we proposed a real-time weed detection method for corn fields which benefits Fast Fourier Transform (FFT) [5] to separate weeds from crops. Although this method has similarities with [4], they have major differences. Firstly, fields of weed detection in these two works are different (a lawn and a corn field), as field-specific parameters such as ground surface shape and field-specific weeds are important for classification purposes. Additionally, in the current work *frequency of edges* are also considered in addition to their densities. Finally, an error correction method is applied to present better results. This algorithm has been tested using images from corn fields showing 92% accuracy. The resulting application is then compiled to a dynamic linked library (dll) and used in a graphical user interface (GUI) to be used further by a cultivator robot in a real field (refer to "Conclusion And Future Works section). Details of the FFT-based algorithm, implemented test application and results, and at last conclusion and future works are presented in section II, section III, and section IV.

II. Fast Fourier Transform-Based Algorithm For Weed Detection

Using specific, distinguishable leaf features of crop and weed for classification is a common method for weed classification (e.g. in [1], [2], [3]). In the proposed algorithm three features are used: Color, frequency of edges, and density of edges. Color differences are used for background removal and edge frequency and density are used for plant classification.

The color segmentation is a fast and accurate method for removing background, preparing the captured image for further processing. The two later features are capable of being used in this classification problem due to special and different orientations of leaf veins in crops and weeds. In leaves of crops like corn (wheat-like leaves), leaf veins are parallel to the main

1-4244-2384-2/08/$20.00

Fig. 1. Amaranth leaves (weed)(left) versus corn leaves (right). Difference between leaf veins strength and orientation, and leaf flatness can be utilized for classification.

vein and the whole leaf has a flat surface. Although this central vein is thick, other leaf veins in the corn leaf is not strong enough to produce strong edges in ordinary edge detection techniques. On the other hand, for main type of corn field weeds, amaranth (pigweed), leaves have an uneven surface, caused by its specific type of leaf veins. Figure 1 shows the difference between leaf shapes of amaranth and corn leaf. This difference cause the edges to appear with a relatively *high frequency* and *high density* in weed leaves which can then detected using proper filters.

On this feature selection basis, the entire algorithm is sectioned into three parts: 1. Pre-processing 2. Frequency and density filtering 3. Post-processing. Pre-processing includes tasks to prepare input RGB format image for filtering processes by removing background, conversion to gray-scale image, and edge detection. After preprocessing phase, filters are applicable to the derived image. Each filter results image segmentation into three regions of crop, weed, and background. At the last step, in order to improve classification accuracy, the post-processing part re-checks the regions for any misclassifications and merges the results into a single image. Weed location in this image are then reported to the cultivator in a matrix format. Figure 2 illustrates flowchart of this algorithm. In this section we describe each of these parts individually using pseudo codes and figures.

A. Pre-Processing

This part of the algorithm prepares an image for further advanced processing and is consists of: Loading the image from source (hard disk or CCD camera), color segmentation, RGB to gray-scale conversion, and edge detection. Due to nature of loading image which is a platform and device dependent task, it is not described in this paper.

In color segmentation part, goal is to differentiate between plants and background with a reasonable accuracy and speed.

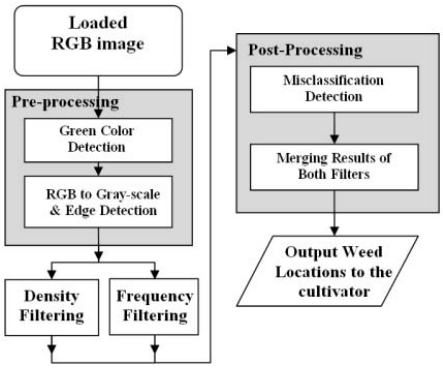

Fig. 2. Algorithm flowchart. Starts with pre-processing, and then filtering. After each filtering, classification errors are detected. Having filter results merged to a single image, weed locations are reported to the robotic cultivator.

Fig. 3. Color segmentation using Euclidean distance with threshold=0.68. Test image from corn field(left); Green parts detected and background is set to black(right).

Although several sophisticated and accurate methods for color segmentation exist, like K-means[7], many of them are not fast enough for our real-time purposes. Therefore we have chosen to use simple Euclidean distance algorithm, applied on green and red values of each pixel for segmentation. According to several tests, blue value of pixels does not affect the segmentation and therefore can be ignored. The Euclidean distance of the pixel to absolute green (R=0, G=1, B=0) is:

$$Pixel\,Color\,Distance = \sqrt{pixel(r)^2 + [pixel(g) - 1]^2} \quad (1)$$

Where $pixel(r)$ is the red value of assessing pixel and $pixel(g)$ is its green value. All pixels having the Euclidean distance more than this threshold were set to black. Figure 3 shows result of this segmentation on a test image.

Final task in the pre-processing is edge detection. In this task the important issue is to mark almost all of leaf veins which normally include a few strong edges and many weak ones. Additionally this process should be fast enough for our real-time problem. As in color segmentation, several methods with different accuracy and speed are available which their most well-known ones are Canny and Soble edge detection algorithms. Although these algorithms can be tuned to find finest edges, their speed reduces drastically in return. As a result, a simple but effective method is applied for edge detection, based on gray intensity changes between two neighbor pixels - after changing the segmented image into gray-scale. If this change is more than a certain threshold, one of the pixels (arbitrarily first or second) is set to white. For better results, this density change tracking should be done in both vertical and horizontal directions. Applying this edge detection method on rows is presented in the following algorithm:

```
FOR all rows in image:
  FOR i from 1 to (row_length-1):
    IF gray_value(pixel_i) -
gray_value(pixel_{i+1}) > edge_Threshold THEN
      set pixel_i=WHITE
    ELSE
      set pixel_i=BLACK
    END IF
  END
END
```

In addition to simplicity and speed, this algorithm can find required edges with arbitrary accuracy without almost any change in its speed. Edge detecting in Fig.4 is done using an experimental threshold equal to 0.05 which showed to be a reasonable threshold in tests. It should be noted that this method of edge detection is reasonable for this work as it is fast and capable of finding edges that is required for frequency filtering. Therefore, it should not be considered as an advanced edge detector algorithm for being used in general purpose works.

B. Frequency And Density Filtering

The pre-processing operations make the input image ready for next step, filtering. At this level frequency and density filters are applied on edge detected plant leaves.

Fast Fourier transform is a well-known and efficient algorithm to compute discrete Fourier transform (DFT) and its inverse. As it is used in solving partial differential equations it is widely used is many applications such as digital signal processing, quantum mechanics, noise reduction, and image reconstruction [9].

Frequency filter is used for recognizing regions of image in which lines are appeared with a frequency in a specific range (weed frequency range). For this to work a grid should be applied on the image and in each block of the grid, a two-dimensional fast Fourier transform (FFT2) should be employed to transform the block to frequency domain. If FFT2 results shows that absolute value of block maximum frequency is in the weed frequency range, the block will be marked as *Weed*; if the frequency is zero, it will be marked as *Background*. The block will be marked as *Crop* otherwise. After this filtering, an $N \times M$ matrix representing estimated locations of crop leaves, weed leaves, and background is available where N and M are the grid dimensions. This matrix is program's actual output to the weeding device informing positions which should be sprayed.

Fig. 4. Edge detection via tracking gray value changes (threshold=0.05)

There is a trade-off between block size and gained accuracy. If the block size is too large, frequency estimation can be faulty due to existence of both crop and weed leaves in a block. If it is too small, the frequency cannot be calculated correctly because of inadequate number of lines in a block. For example a small block may mark an inner part of a weed leaf as a crop because there is only one or two edges included in block area. Additionally, the thresholds confining the weed frequency range should be set to classify the leaves optimally.

Although block size can be estimated experimentally, the thresholds accuracy should be calculated carefully to present optimal classification results. A linear classifier such as Fisher linear classifier (FLD) [8] can be used for this calculation. The result from the FLD is then used as a threshold to differentiate between weed frequency range and crop frequency range.

Density filter is similar to frequency filter as it crops the image into blocks and tries to differentiate between crop leaves and weed leaves using their edge densities. This is again feasible because detected edges for weed leaves are noticeably more than crop leaves. Leaf shapes in Fig.1 and detected edges in Fig.4 show this density variation.

Having edge detection output image, the density filter section divides the image into appropriate blocks and counts white pixels in each block. The blocks are again categorized into three types: 1) Background, having no or very few white pixels; 2) Weed, having more than a certain number of white pixels; and 3) Crop, the only other case where the number

C	C	C	B	B
C	C	C	C	B
C	C	**W**	C	C
B	C	W	W	C
B	C	C	C	C

Fig. 5. A block surrounded by opposite-type blocks which will be considered as a misclassification. C, W, and B represent crop type, weed type, and background type respectively. Center block (having bold W letter) is the block being check.

of white pixels are between the minimum threshold and weed range threshold. Although Same block size as frequency filter block size can be used for this part (which facilitates merging matrices), a new linear classifier should be employed to find the threshold optimally. Here again, FLD can be used as well.

C. Post-Processing

Although frequency and density features can be tuned for an optimal classification, there will be probability still false classifications. Therefore another part is added to presented weed detection algorithm as a final correction section. This post-processing part is based on the fact that a block of one leaf-class cannot be surrounded by blocks of the other leaf-class, where leaf-classes are either weed or crop and the block size is small enough. In other words, if the algorithm has detected a block of weed (crop) in a crop (weed) leaf, it is highly probable that a misclassification has been occurred.

To apply this constraint, a cell-by-cell check is performed. For each cell, number of opposite-class surrounding cells are counted. If this number is more than a certain threshold, the cell label will be toggled from one leaf type to the other. Surrounding cells are defined as the 24 cells around each cell (except border cells). Figure 5 shows a cell surrounded by opposite type cells. Although this method seems simple but tests showed that it can eliminate more than 50% of false detections.

After rechecking, two $N \times M$ matrices are available, in which each cell represents a block of original image, marked by either *crop*, *weed*, or *background*. After performing the post-processing task, these two matrices should be collapsed into a single matrix using Boolean *AND* function on weed-marked cells. It means if at least one of the frequency matrix or density matrix cells is marked as weed, the related union cell in the resulting matrix will be marked as weed. This leads to more accurate detection of weed leaves using both frequency and density filters results. The other two flags, crop and background, are no more important because cultivator device merely consider the weed blocks. As a result an $N \times M$ matrix is returned from the entire algorithm.

III. Test Implementation and Results

In this section details of a test based on the presented algorithm are described. MatLab 7.0 is used for this implementation and finally a dynamic linked library (dll) was compiled and used in a C#.Net based graphical user interface (GUI) to be used easily in future works.

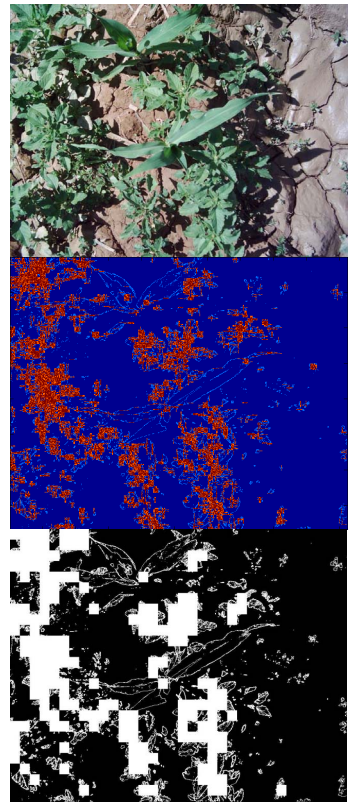

Fig. 6. Sample result of applying frequency and density filters to detect weed regions: Original image (up); Frequency filter result (bottom-left); Density filter results (bottom-right).

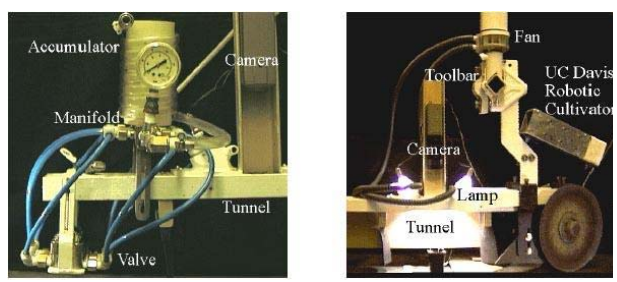

Fig. 7. UC Davice robotic cultivator: (Right) Artificial lights and Camera configuration. (Left) Spraying component

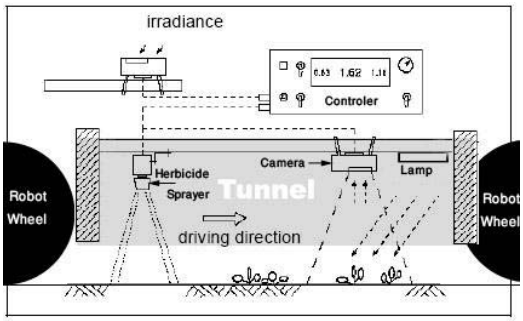

Fig. 8. Cross cut of cultivator tunnel. Showing lighting system, camera, sprayer, irradiance sensor, and digital controller

After implementing the algorithm in MatLab code, 80 corn field images from field images database of Agricultural School of Shiraz University have been used to test the classification accuracy. The images were captured in the noon sunlight and half of them were re-sized to 640×480 pixels and the other half to 1024×768 pixels. These images were stored on a hard disk and were loaded from there instead of a digital camera.

The tests were executed on a 3GHz Pentium 4 Intel processor using 512MB of main memory. Average times were 0.2 sec. for small size images and 0.5 sec. for larger images. After applying the GUI the average time increased to 0.3 sec. and 0.7 sec. respectively due to GUI overhead.

For green color detection section, threshold was set to 0.68. This value showed the best performance for removing background in our empirical results. Edge detection used the method described in section 2, with its threshold set to 0.05.

Applying filters, two-dimensional fast Fourier transform (FFT2) and density, were the second part of the algorithm which were done having 15×15 pixel blocks. For the FFT2, two-dimensional Cooley-Turkey algorithm [10] was used. Manually labeled (weed/crop) absolute values of the transformations from 10 images were used as input data to an FLD for calculating a classification threshold which was 78.9.

A same procedure was followed for density filter with a small difference. For this filter, the threshold was defined relative to block size to alleviate block size changing. Block density data (number of white pixels in a block) from 10 manually labeled images were used for FLD training resulted in a density threshold equal to 30.50% of the block size. One sample from each filtering operations has been presented in Figures 6.

For calculating algorithm accuracy, the following formula was used:

$$Accuracy = \frac{CC}{WD + CD}$$

where CC is number of blocks which are correctly classified, WD is number of blocks which are detected as weed, and CD is number of blocks detected as crop. Algorithm's average accuracy over all 80 images was 92.8%. Table I illustrates some of our records from 1024×768 images.

	WD	CD	CC	%
1	2015	1007	2717	89.9
2	1302	597	1859	93.5
3	1961	740	2526	93.5
4	649	873	1477	94.8
Total	5927	3217	8579	93.8

TABLE I

SOME OF THE ALGORITHM RESULTS. WD: NUMBER OF BLOCKS DETECTED AS WEED; CD: NUMBER OF BLOCKS DETECTED AS CROP; CC: NUMBER OF CORRECT CLASSIFIED BLOCKS.

IV. CONCLUSION AND FUTURE WORK

This algorithm will be used in a weeding robot, built in Mechanical Department of Shiraz University, for being tested in an actual corn field. For increasing detection accuracy, artificial illumination is considered. One of the candidates for weeding robot architecture is the cultivator presented by Lee [6]. Figures 7 and 8 shows different components of this model and their configurations.

REFERENCES

[1] J. Hemming and T. Rath, "Computer-vision-based weed identification under field conditions using controlled lighting," Journal of Agricultural Engineering Research, vol. 78, no. 3, pp. 233-243, 2001.
[2] W.S. Lee, D.C. Slaughter, and D. K. Giles, "Development of a machine vision system for weed control using precision chemical application," Proceedings of ICAME-96, vol. 3, pp. 802-811, 1996.
[3] W.S. Lee, D.C. Slaughter, and D.K. Giles, "Robotic weed control system for tomatoes," Precision Agriculture, vol. 1(1), pp. 95-113, 1999.
[4] Watchareeruetai, U.; Takeuchi, Y.; Matsumoto, T.; Kudo, H.; Ohnishi, N., "Computer Vision Based Methods for Detecting Weeds in Lawns," Cybernetics and Intelligent Systems, 2006 IEEE Conference on , vol., no., pp.1-6, June 2006
[5] Glassman, J.A., "A Generalization of the Fast Fourier Transform; Computers," IEEE Transactions on Volume C-19, Issue 2, Page(s):105 - 116, Feb. 1970
[6] Development of A Machine Vision System For Weed Control Using Precision Chemical Application; Won Suk Lee et al.; Department of Biological and Agricultural Engineering, University of California, Davis, Davis, CA95616, USA
[7] Reference for K-Means
[8] Fisher, R.A. The Use of Multiple Measurements in Taxonomic Problems. Annals of Eugenics, 7: 179-188, 1936
[9] Oraintara, S.; Chen, Y.J.; Nguyen, T.Q., "Integer fast Fourier transform"; IEEE Transactions on Signal Processing, Volume 50, Issue 3, Page(s):607 - 618, March 2002
[10] Cooley, James W., and John W. Tukey, "An algorithm for the machine calculation of complex Fourier series," Math. Comput. 19: 297–301, 1965