Application of Image Processing Techniques in the Characterization of Plant Leafs

José BOAVENTURA CUNHA

UTAD, Dep. Engenharias, 5001-911 Vila Real, Portugal, e-mail : jboavent@utad.pt

Abstract— **The number of applications using machine vision and digital image processing techniques in the agricultural sector is increasing rapidly. These applications include land/aerial remote sensing of crops, detection and recognition of pathological stress conditions, shape and color characterization of fruits, among many other topics. In fact, quantification of the visual properties of horticultural products and plants can play an important role to improve and automate agricultural management tasks. In this paper, is described a plant leaf characterization system based on a personal computer. This system uses a digital scanner to acquire leaf images with a resolution of 150dpi. These images are afterwards processed in order to compute some leaf characteristic parameters, such as: leaf area and perimeter, existence of holes, width and length. With the implemented algorithms the errors between the measurements and the real values were typically less than ±3% and ±2.5% for the area and linear measurements, respectively. These tests and results were realized using sets of known size images and leaf images that were measured with the proposed system and with a commercial calibrated leaf area system *LiCor* from Delta-T Devices.**

Index Terms— **Digital Image Processing, Edge Algorithms, Leaf Area Index.**

I. INTRODUCTION

Nowadays, one of the most important tools that are being used to control and manage the horticultural production is the crop model. Since the present sensing technologies can not be applied in a practical way to measure the physiological response of the plants, these models are essential to optimize the production plan [1], [2]. Generally speaking, dynamic simulation crop models are based on the description of the crop photosynthesis and respiration processes with emphasis on dry matter and leaf area production. In order to have accurate models, periodic destructive tests must be conducted to measure plant leaf area. Here, is described a low-cost system, based on a personal computer and a digital scanner from Hewlett Packard, to perform plant leaf characterization.

II. MATERIALS AND METHODS

In this section are described the software tools that were implemented to perform the measurement of the plants leafs. First, a set of leaf images taken from one or more plants are acquired with a resolution of 150dpi and, afterwards, they are subjected to image processing techniques in order to compute the area, perimeter, existence of holes, width and length. The image processing algorithms were implemented using the software package *MATLAB* from MathWorks. Since the digital scanner has an image acquisition area of 22cm by 30cm, it was implemented an integrating mechanism to measure the leaf parameters for the all plant or set of plants.

The acquiring and processing image software enables the visualization and storage of color images in the file format *JPEG* – Joint Photographic Experts Group. Prior the start of the estimation of the leaf characteristics, several image processing techniques are applied to the original image, such as: Color to grayscale conversion, histogram equalization, noise reduction, contrast enhancement and binary conversion. In this way, several images are generated with the purpose of being used within the different digital processing image techniques implemented.

Some of these image processing techniques are single point operations that are used for image enhancement. This process can be described by a mapping function $P_o=M(P_i)$ where P_o and P_i denote the pixel values in the output and input image. The function M can be defined in an ad-hoc manner, as for the image thresholding or can be computed from the input image as in the case of histogram equalization. For example, a simple threshold operator is defined by the mapping function:

$$P_o=\begin{cases}0, & P_i>T\\ L-1, & P_i\le T\end{cases}, \qquad (1)$$

where: T defines the threshold level of the gray level transformation and L is the gray level range.

In this case the input to this operator is a grayscale or a color image. Also, it is possible to specify two different level thresholds. This simple image processing technique is useful to generate a binary image representing the segmentation, where black pixels correspond to foreground and white pixels to background. The resulting binary image is used to compute the leaf area as the sum of the pixels set to zero (black). Also, this image is applied to other image processing operators, such as the Prewitt edge detector in order to compute the leaf perimeter.

The threshold level used to generate the leaf binary image representing the segmentation, can be manually adjusted with a slider until the result is satisfactory. Also, it were implemented techniques to automatically choose a threshold starting from the gray-value histogram,

0-7803-7912-8/03/$17.00

$\{h[b] \mid b = 0, 1, \ldots, 2^B\text{-}1\}.$ (2)

These algorithms can benefit from a smoothing of the raw histogram data to remove small fluctuations, but this must be done without shifting the peak positions. The following zero-phase smoothing algorithm could be employed to perform this operation.

$$h_{smooth}(b) = \frac{1}{Nr} \sum_{n=(-Nr+1)/2}^{(Nr-1)/2} h_{raw}(b-n), \quad Nr \text{ odd}, \tag{3}$$

where typical values for Nr are 3 or 5.

Pixels below the threshold $p(m,n) < \theta$ will be labeled as object pixels and the remaining will be labeled as background pixels.

Also, an iterative technique for choosing a threshold was proposed by Ridler and Calvard [3]. The histogram is initially segmented into two parts using a starting threshold value such as $\theta_i = 2^{B-1}$, half the maximum dynamic range. The sample mean $(m_{f,i})$ of the gray values associated with the foreground pixels and the sample mean $(m_{b,i})$ of the gray values associated with the background pixels are computed. A new threshold value θ_i is computed as the average of these two sample means. This process is repeated according to:

$$\begin{aligned} &while\ \ \theta(k) \neq \theta(k-1) \\ &\quad \text{do,}\ \ \theta(k) = (m_{f,k-1} + m_{b,k-1})/2 \end{aligned} \tag{4}$$

Noise removal is a crucial issue in order to achieve a reliable measurement system based on data image processing. In this work a median filter is used to remove outliers since it has the feature of preserving the sharpness of the image [4]. This is due to the fact that the filtered output pixel is determined by the median of the neighborhood pixels instead of their mean value. In the current implementation the size of neighborhood pixels used for filtering can be selected between 3x3 and 5x5.

Next figure illustrates the process for computing the median filtered image using a 3x3 square neighborhood. If the neighborhood selected contains an even number of pixels, the average of the two middle pixels is used.

122	123	126
120	151	144
115	129	119

Algorithm:

Compute the pixel neighborhood sorted values:
115, 119, 120, 122, **123**, 126, 129, 144, 151

Compute the median value: 123

Substitute the central pixel value of 151 with the median value 123.

Fig.1 Operation of the median filter of size(3x3)

To illustrate the benefits of applying the median filter, next figures shows the binary images generated from the threshold operation applied to the leaf gray level image of Fig. 2. In the image of Fig. 3 can be seen the presence of noise and in the binary image of Fig.4 the noise was removed due to the use of the median filter.

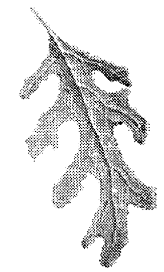

Fig.2 Acquired image of a tomato leaf after conversion to gray level

Fig.3 Threshold of the gray level image showing some noise

Fig.4 Threshold of the image after applying a (3x3) median filter

The image of Fig. 2 results from the color to gray level conversion of an original leaf image. If this image is used to compute a binary image by thresholding some noise appears around the leaf binary image, Fig.3.

This will degrade the edge computation and will lead to erroneous measurements on the leaf area and perimeter. To avoid this problem a median filter is applied to the gray level image and then the thresholding performed originating the image of Fig 4.

The user can carry the leaf measurements using a friendly interface, either in manual or automatic mode. In the manual operation the operator can select image regions or points to compute the areas or distances. In the automated operation the area and sizes are computed for all the leafs detected.

Each leaf area L_i is computed as the number of points present in that segmentation set. Leaf perimeter is estimated by performing edge detection of the leaf image. Several methods could be selected for this purpose such as the Prewitt, Sobel, and Laplacian algorithms [5], [6].

In this case, it was implemented the Prewitt method since it has proofed to be well suited to extract the image edges regardless of direction.

Assuming that the image is a full scan of a single leaf, the Prewitt edge detector is used to produce an image that emphasizes the regions of high spatial frequency. This edge algorithm computes the root mean square of two 3x3 kernels, showed in (5), that are choose to respond maximally to edges running vertically and horizontally relative to the pixel grid [7].

$$Gx = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, \quad Gy = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}. \qquad (5)$$

These kernels are applied to the image to produce the measurements of the gradient component in each direction and, afterwards, they are combined to compute the Prewitt gradient (*PG*), as described below.

Considering the image window (6),

$$IW = \begin{bmatrix} P_1 & P_2 & P_3 \\ P_4 & P_5 & P_6 \\ P_7 & P_8 & P_9 \end{bmatrix}, \qquad (6)$$

where P_1 to P_9 are the gray levels of each pixel in the image window.

The Prewitt gradient magnitude is computed employing the equations (7.1 to 7.3):

$$Gx = -1 \times P_1 + P_3 - 1 \times P_4 + P_6 - 1 \times P_7 + P_9, \qquad (7.1)$$

$$Gy = P_1 + P_2 + P_3 - 1 \times P_7 - 1 \times P_8 - 1 \times P_9, \qquad (7.2)$$

$$PG = \sqrt{Gx^2 + Gy^2}, \qquad (7.3)$$

Normally, the edges produced by the Prewitt convolution kernel are wider than one pixel. To avoid this effect it is necessary to employ a thinning algorithm.

The thinning operator must reduce the detected edge lines to lines of a single pixel width and must preserve the full length of those lines.

There are two basic approaches to edge thinning. One method involves an interactively application of sets of masks to delete unwanted pixels and the second method uses logical and arithmetic operations for contour tracing [8], [9].

A simple algorithm that can be used to perform the edge thinning is illustrated in Fig. 5.

This algorithm erodes away the boundaries of foreground objects without affecting pixels at the end of lines.

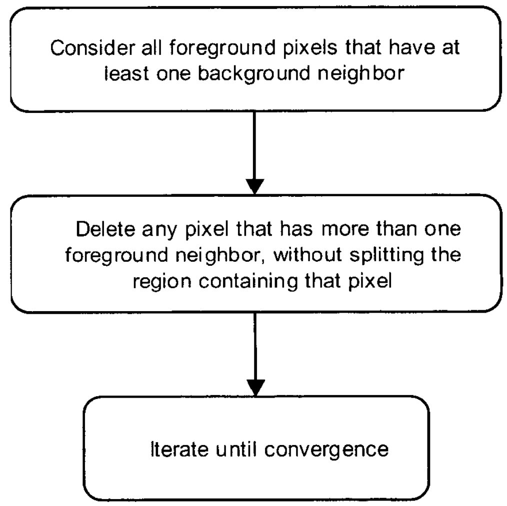

Fig.5 Operation of the median filter of size(3x3).

This operation can be obtained using morphological thinning by iterating until convergence with the kernel structures shown in Fig. 6. and all their 90° rotation kernels, which corresponds to a total 8 structuring kernels

0	0	0
	1	
1	1	1

	0	0
1	1	0
	1	

Fig.6 Structuring kernels for skeletonization by morphological thinning.

This procedure consists in the determination of the octagonal skeleton of a binary shape, i.e., the set of points that lie at the centers of octagons that fit inside the shape, and which touch its boundary in at least two points. This skeletonization method always produce a connected skeleton.

At each iteration, the image is first thinned by the left hand structuring kernel, and then by the right hand one. Afterwards, this operation is repeated for the remaining six 90° kernel rotations. The process is done cyclically until no more changes are produced by the thinning operator

In the next figure is illustrated the result of this thinning operation over a binary image.

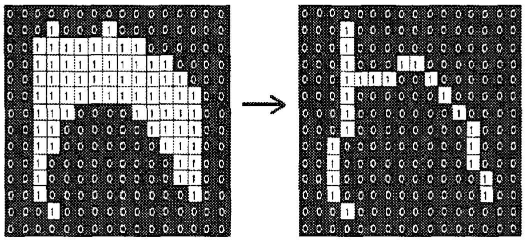

Fig.7 Skeletonization by morphological thinning of a binary shape.

The skeletons produced by this method often contain undesirable short spurs produced by small irregularities in the boundary of the original object. These spurs can be removed by a process called pruning, which is another sort of thinning. Pruning is normally carried out for a limited number of iterations to remove short spurs. This is due to the fact that pruning until convergence will remove all pixels except those that form closed loops.

III. RESULTS

The software algorithms described in the previous sections were implemented in a friendly user interface, showed in Fig. 8. The user interface show the acquired leaf image, the threshold image after applying the median filter and the leaf contour after applying the Prewitt edg detection algorithm. The contour white pixels are used t derive the linear dimensions of the leaf. For instance, tl perimeter is computed as the sum of the number of whi pixels that constitute the contour.

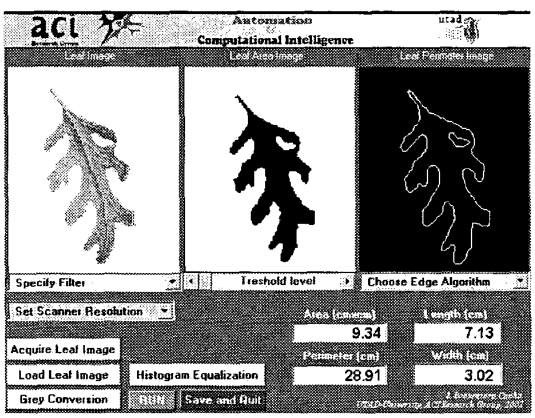

Fig. 8 Leaf image processing menu: acquired image of a tomato leaf (left), threshold of the image after applying a (3x3) median to reduce the noise (middle) and edge computation (right).

The leaf measurements can be computed either in manual or automatic mode.

In the manual operation the operator can select image regions or points to compute areas and linear distances. Also, by using pop-up menus and sliders it can be selected various digital image processing algorithms. For instance, the threshold level used to generate the leaf binary image representing the segmentation, can be manually adjusted with a slider until the result is satisfactory.

In the automated operation the area and sizes are computed for all the leafs detected in an acquired or loaded color image. In this case, the filter, threshold level and edge detection algorithms are tuned in an adaptive way. For instance, to auto-tune the threshold it were implemented techniques that automatically choose a threshold starting with the gray-value histogram.

With the implemented algorithms the errors between the measurements and the real values were typically less than ±3% and ±2.5% for the area and linear measurements, respectively. These tests and results were realized using sets of known size images and leaf images that were measured with the proposed system and with a commercial calibrated leaf area system *LiCor* 3000A from Delta-T Devices.

Next figures show some of the results achieved for the area measurement of a set of predefined shapes (squares, triangles, circles, etc.) with known area, Fig.9, and the ones obtained by comparing the measurements of real leafs computed using the proposed hardware and algorithms, with the measurements performed with the *LICOR* area meter from Delta-T Devices, Fig.10.

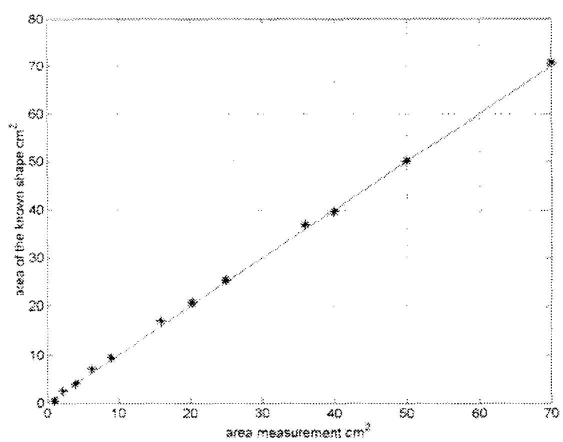

Fig.9 Ideal response (-) and comparison between area measurements and real values (*)

From Fig.9 it can be seen that the areas computed with this software package are very close to the known real values.

Also, the measurements made with a precision calibrated leaf area meter *LICOR 3000A* and the measurements computed with the proposed system, plotted in Fig.10, are very closely related.

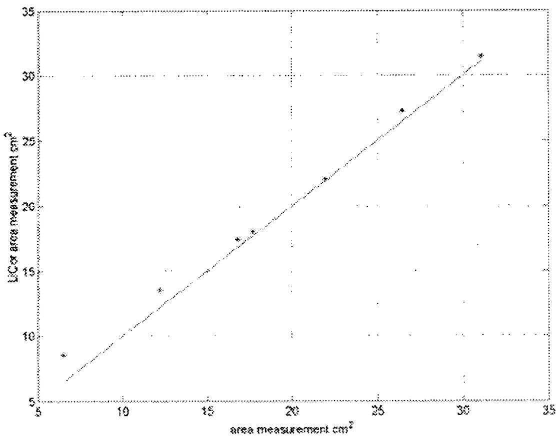

Fig.10 Comparison between area measurements of the proposed system and the LiCor area meter instrument (*)

IV. CONCLUSIONS

In this work were described digital image processing techniques to characterize plant leafs. The implemented algorithms proofed to be adequate to compute the leaf areas and sizes, which are essential to calibrate plant growth models, since their responses are affected by the environmental conditions and by the *LAI*-Leaf Area Index.

The proposed tools do not able to perform the calibration process of the growth models automatically, since it is needed to collect plants in site. However, it must be referred that, in common practice this process is done once a weak to calibrate the sub-model of *LAI*.

At the present other experiments are being conduced to detect the presence of holes and changes in leaf color in order to infer about the presence of diseases. This component will be valuable to achieve a better estimate of the plant photosynthesis response and so of the growth response of the plant.

V. REFERENCES

[1] J. Boaventura Cunha, P. Oliveira and M. Cordeiro, " Optimal Control of Air Temperature and Carbon Dioxide Concentration in Greenhouses", in *Proceedings. of the World Congress of Computers in Agriculture and Natural Resources*, March 13-15, 2002, Iguaçu Falls, Brazil, pp:168-175.

[2] V. Fernandes and J. Boaventura Cunha, "Evaluation of Plant Growth Models for a Soil Greenhouse Tomato Crop", in *Proceedings of the World Congress of Computers in Agriculture and Natural Resources*, March 13-15, 2002, Iguaçu Falls, Brazil, pp:147-153.

[3] Ridler, T.W. and S. Calvard, "Picture thresholding using an iterative selection method". *IEEE Transactions on Systems, Man, and Cybernetics*, 1978. SMC-8(8): pp: 630-632

[4] M. Heath, S. Sarkar, T. Sanocki and K.W.Bowyer, "A Robust Visual Method for accessing the Relative Performance of Edge-Detection Algorithms",. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 12, December 1997, pp:1338-1359.

[5] John C. Russ. *The Image Processing Handbook, fourth edition*, Boca Raton, Florida, CRC Press, 2002, 250pp.

[6] Gregory A. Baxes. *Digital Image Processing*. New York, John Wiley and Sons, 1994, 290pp.

[7] Ioannis Pitas. *Parallel Algorithms for Digital Image Processing, Computer Vision and Neural Networks*, Wiley and Sons, 1993, 320pp.

[8] H. K. Aghajan, C.D. Schaper and T. Kailath, "Edge detection for optical image metrology using unsupervised neural network learning", in *Proceedings of IEEE Workshop on Neural Networks for signal Processing*, 1991, pp:188-197.

[9] H. K. Aghajan and T. Kailath, "A subspace fitting approach to super resolution multi-line fitting and straight edege detection", in *IEEE Acoustics, speech and Signal Processing*, 1992, pp:188-197.