LEAF SEGMENTATION AND PARALLEL PHENOTYPING FOR THE ANALYSIS OF GENE NETWORKS IN PLANTS

Olivier Janssens[1,2], *Jonas De Vylder*[3], *Jan Aelterman*[3], *Steven Verstockt*[2]
Wilfried Philips[3], *Dominique Van Der Straeten*[4], *Sofie Van Hoecke*[1,2], *Rik Van de Walle*[2]

[1]Electronics and Information Technology lab, ISP, Ghent University
Graaf Karel de Goedelaan 5, 8500 Kortrijk, Belgium
Telephone: + 32 56 24 12 52; email: odjansse.janssens@ugent.be

[2]Multimedia Lab, ELIS, Ghent University – iMinds

[3]Dept. of Telecommunications and Information Processing, TELIN, Ghent University – iMinds

[4]Laboratory of Functional Plant Biology, Department of Physiology, Ghent University

ABSTRACT

Over the last 4 years phenotyping is becoming more and more automated, decreasing a lot of manual labour. Features, which uniquely define the plant, can be extracted automatically from images. As a lot of plant data has to be processed in order to extract the features, fast processing of these features is a challenge. Therefore in this paper, a new method for automatic segmentation of individual leaves from plants with a circular arrangement of leaves (rosettes) is proposed, together with an algorithm to extract the line of symmetry of the leaf. Furthermore, in order to achieve fast processing for phenotyping plants, four feature extraction methods are parallelised in order to run on the CPU and GPU. Our evaluation results show that by parallelizing the feature extraction methods, it is possible to calculate the image moments, area, histogram and sum of intensities 5 to 45 times faster than single threaded implementations.

***Index Terms*—** segmentation, parallelisation, phenotyping, OpenCl, image processing

1. INTRODUCTION

Plant analysis by computer vision is a promising non-destructive method which can be performed in an automatic way. Several methods have been proposed in literature which focus on the leaves solely, such as the LIMANI [1] framework or LEAF GUI [2]. The LIMANI framework focuses on the automatic segmentation and measurement of venation patterns. LEAF GUI is a user-assisted software tool that facilitates improved empirical understanding of leaf network structure. These frameworks are of value for analysing plants, though the disadvantage is the fact that leaves are not segmented automatically. Apart from the venation patterns it is also possible to extract other features such as plant growth [3], area of the leaves, relative growth rate, compactness of the rosette, diameter of the rosette, stockiness, and intensity of the plant image [4–7]. The biggest limitation, however, is that these features are usually not extracted from the leaves solely, but rather from the entire rosette. All these features have to be calculated for each plant and for every time frame, requiring a lot of processing time, which can introduce a bottleneck in the analysis of plants. Because of the before mentioned problems, we can state there is a need for (1) automatic leaf segmentation (i.e., in the discussed state-of-the-art leaf segmentation is done manually) and (2) fast processing for feature extraction to cope with growing datasets. Therefore, an algorithm capable of segmenting individual leaves is presented in this paper. Additionally, new and generic features are proposed that allow for parallel processing in order to achieve fast processing of the plants.

The remainder of this paper is as follows: Section 2 explains how automatic leaf segmentation from a plant rosette can be done. Subsequently section 3 introduces the algorithm to extract the line of symmetry from a leaf. Next, Section 4 gives an overview of features that are extracted in parallel from the rosette. Evaluation results are presented in Section 5. Finally, Section 6 ends this paper with conclusions.

2. LEAF SEGMENTATION

In order to be able to extract features accurately, it is necessary to segment the rosette as precise as possible. As described by De Vylder et al. [7], several effective and efficient methods exists to extract individual rosettes such as supervised pixel classification methods, pixel clustering based methods and threshold methods. In this paper, chlorophyll fluorescence imaging is used in order to capture images of the plants. As can be seen in Figure 1 a), an image taken with chlorophyll fluorescence imaging results in a grayscale image which is easy to segment by a fixed threshold. This is done by comparing every pixel value to an empirically determined threshold value in order to decide if the pixel belongs to the plant or the background. By using a fixed threshold, more complex and computational demanding methods are avoided [7].

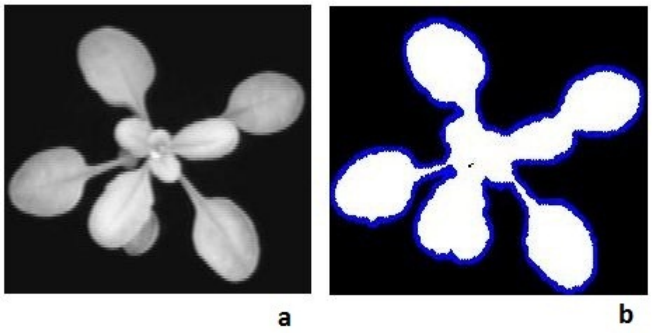

Fig. 1. a) Chlorophyll fluorescence image of a plant. b)Image and contour of plant after applying thresholded segmentation.

The advantage of using a fixed threshold is the simplicity of the implementation. Using a fixed threshold, however, is only possible if the images are similar to each other. In order to guarantee similar illumination for the plants, we need to work in a controlled environment, which is the case for chlorophyll fluorescence imaging.

After the threshold segmentation of the rosette, the leaves needs to be segmented from the plant. In order to do so, the contour from the segmentation is extracted first. The contour is a useful feature that helps to indicate where the leaf starts and ends.

2.1. Leaf extraction

In order to extract the contour from a binary image, the Moore neighbourhood algorithm is used [8]. The Moore neighbourhood algorithm uses an image where the background is segmented from the foreground and runs over the pixels on the edge of the foreground. An example of an extracted contour can be seen in Figure 1 b). Using the found contour, segmentation is done in three steps.

The **first step** consists of finding the pixels of the contour which belong to the stem, since those are indicative of where the leaf starts and the stem ends. Our algorithm takes advantage of the fact that pixels of the stem lie close to each other in the image, but are not close together in the list of pixels belonging to the contour. In order to find these stem pixels, every pixel of the contour is compared to every other pixel of the contour. During this comparison, two distances are calculated: the Manhattan distance between the two pixels and the distance between the two pixels in the list of the contour pixels. The Manhattan distance, a fast metric without multiplications or powers, is a relatively small number as two pixels of the stem should be close to each other. The second distance is the difference of the absolute value of the indices of the pixels in the contour list. This distance value is bigger since two pixels of the same stem are separated by the contour pixels of the leaf. An example of points marked by the algorithm can be found in the first step of Figure 2.

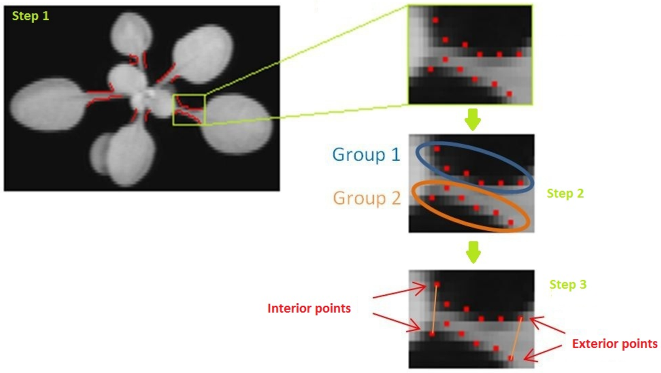

Fig. 2. In order to segment the stems, the pixels from the contour belonging to the stem are found. Secondly, consecutive points are put in a group together. Finally the first and last points of the groups are connected

As can be seen, points come in groups. Therefore, **step 2** of our algorithm assigns consecutive points on the contour to the same group; per stem of a leaf, there are two groups (see Step 2 Figure 2).

In order to segment the stem, the interior (i.e. the points closest to the rosette) and exterior points (i.e. the points furthest away from the rosette) of the two groups are connected. These connections are determined by following the contour (see Step 3 Figure 2). The resulting two lines indicate where the leaf starts and the stem ends. By combining these lines with the contour of the rosette, the leaf can be segmented accurately.

Some examples of segmented leaves can be seen in Figure 3. In these images, leaves 1, 2 and 4 are segmented clearly, but leaf 3 is in fact two leaves overlapping. This problem only occurs when the rosette has a lot of leaves and the leaves start to overlap, which is not the case when the plant is still young. If the leaves are not overlapping, a clear segmentation of the individual leaves will happen.

As the leaves are now segmented, it is possible to extract features for every individual leaf.

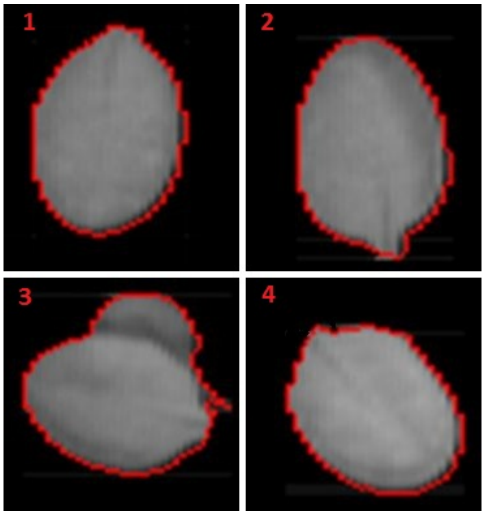

Fig. 3. Leaves that are segmented from the rosette.

3. LINE OF SYMMETRY EXTRACTION FROM INDIVIDUAL LEAVES

One of the main features that can be calculated based on a leaf solely, is the line of symmetry. The line of symmetry can help to determine how symmetric or asymmetric a leaf is. This information can help to link this feature to environmental conditions or disease symptoms such as lesions.
As is shown in Figure 4, the line of symmetry of a leaf can be extracted by the following three steps:

1. Segment the leaf and the stem.
2. Calculate the centroid based on the image moments of the leaf and the stem.
3. Draw a line between the centroid of the leaf and the centroid of the stem.

The first step is explained in the previous section where the stem is segmented in order to get the leaf. The second step starts by calculating the image moments based on the contours. Image moments are statistical features, which can give properties such as the skewness, kurtosis, area and perimeter. Image moments are calculated following Eq. 1, where $i+j$ defines the order of the image moment. In this equation $I(x,y)$ is the value at position x and y of the contour.

$$M_{i,j} = \sum_x \sum_y x^i y^j I(x,y) \qquad (1)$$

Using these moments it is possible to calculate the centroid of the stem and the leaf based on Eq. 2. Both centroids can be seen in Figure 4 at step 2. In Eq. 2, $\bar{x}$ is the x coordinate of the centroid and $\bar{y}$ is the y coordinate of the centroid. Since the 2 centroids are known, the line of symmetry can be drawn between them.
Based on this line of symmetry it is then, for example possible to divide the area of the leaf in two and see how much the two parts of the leaf differ.

$$\bar{x} = \frac{M_{10}}{M_{00}}; \bar{y} = \frac{M_{01}}{M_{00}} \qquad (2)$$

The line of symmetry is only one of the many features which can be extracted from the leaf. In the next section we discuss 4 other commonly used features, i.e. the image moments, area, histogram and the sum of intensities, from the rosette are discussed. They are calculated in parallel in order to speedup the feature calculations.

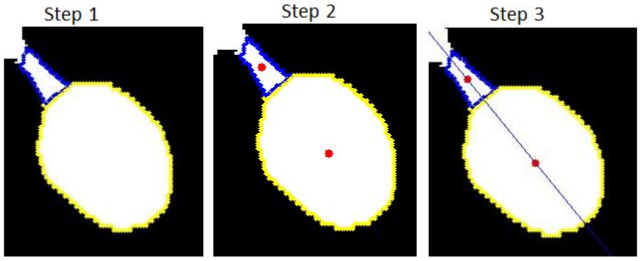

Fig. 4. The first step in order to get the line of symmetry consist of segmenting the leaf and the stem. In the second step the centroids are found and in the 3th step the centroids are connected by the line of symmetry.

4. PARALLEL PROCESSING OF LEAF FEATURES

In order to provide a solution for the need of fast processing, which is introduced in Section 1, the calculation of phenotyping features are parallelised to achieve fast processing. In our work, the decision is made to extract four relevant phenotyping features namely (i) image moments based on the contour of the rosette; (ii) the area - since it is indicative of the plant growth; (iii) the histogram - because it is a widely used representation of an image and; (iv) the sum of intensities. The sum of intensities of an image is also used as a feature in plant phenotyping since the fluorescence emission by the chlorophyll, which can be seen in the in the intensity values, indicates how efficient the plant utilises photosynthesis.
In order to speed up the calculations of these features, multiple images are processed in parallel. This kind of parallelism is called data-parallelism, which means that the same instructions are applied on different sets of data at the same time. This is done with loop unrolling, where the amount of images is divided in equivalent sets. On every iteration, several images are processed in parallel.

We have chosen this kind of parallelism methodology because it results in straight forward calculations. Another approach is for example subdividing the images in smaller images so that more calculations can be done in parallel. The latter would cause more overhead as images only have a resolution of 300 x 300 pixels.

The next section discusses the results of the first methodology using loop unrolling.

5. RESULTS

Results of the leaf segmentation and symmetry calculation algorithms are presented in this section. Subsequently the results of the parallelisation of the feature calculation methods are discussed.

5.1. Leaf segmentation and line of symmetry calculation

Regarding plant analysis, it can be stated that a subjective analysis indicates that our algorithms clearly segments the leaves and calculate the line of symmetry. The segmented leaves from the plants (such as in Figure 3) were presented to experts in the field of rosette phenotyping and judged that the leaves are clearly segmented and are suitable to work with.

5.2. Parallel processing of leaf features

In order to test the speedup of the data-parallelism approach for feature calculations, every calculation is done multiple times in order to eliminate possible outliers. In this research every calculation is done a 100 times on 5120 images with a resolution of 300 x 300 pixels. All the calculations were done both sequentially and parallel on the CPU and parallel on the GPU. The devices used for these benchmarks are the intel i7 870 with 8 logical cores and the ATI Radeon HD 5870 which consists of 20 computing units, each consisting of 80 processing elements.
The OpenCl framework is used to parallellize the feature extraction methods as OpenCl allows the same code to run on both the CPU and GPU. In OpenCl the code that has to be executed on the CPU or GPU is defined in the Kernel. A work-item (i.e. a thread) is a single implementation of the kernel and is able to access the resources of the work-group. All the work-items in a work-group can access the same resources. In this case the resources per work-group consist of a range of images. Each work-group is then assigned to a computing unit. To understand how OpenCl works, two cases are presented:

Case 1

> 5120 work-groups are created with each work-group consisting of 1 work-item, which is defined by the localsize. In our GPU there are 20 computing units, which means that all computing units are occupied, though not all the processing elements are working. This is because the localsize is set to one, which means there is only one thread in each work-group.

Case 2

> 20 work-groups are created with each 256 work-items. Each computing unit is occupied and has 256 images to process each. Since there are 80 processing items per computing unit all of them are occupied processing an image. When a work-item is done the scheduler swaps the work-items so that all the work-items (and therefore the images) get processed.

In order to speedup the results, experiments are done with different sizes of work-groups. The results for the four features i.e moments, area, histogram and sum of intensities calculations can be seen in Table 1 for the GPU & Table 2 for the CPU.

When looking at the results for the image moment and area calculations it can be stated that the GPU performs much better than the CPU. This can be explained by the fact that both of these functions require a lot of calculations in comparison to the amount of memory accesses. As more time is spent to calculating in comparison to the time spent to reading and writing of the data, this is an advantage for the GPU that it designed to do many calculations and is not designed for a lot of memory accesses.
As can also be seen in Table 1 and Table 2, calculating the histogram of an image involves extracting the addresses from the data, resulting in extra memory access and a reduced speedup, especially on the GPU. On the other hand, as can be seen for the area calculations, when the write address is known in advance, the calculations are a lot faster. As a result it can be said that the CPU is a better choice when it comes to achieving fast processing for histogram calculations.
The results for the intensity calculations presented in both tables confirm the advantage of GPU over CPU for speeding up feature calculations.

6. CONCLUSION AND FUTURE WORK

In this paper we show that it is possible to automate leaf segmentation from rosettes. The proposed algorithm gives rise to the possibility to automate leaf analysis tools even further, since the current state-of-the-art leaf analysis tools require to manually segment the leafs. By automatically segmenting leaves, less time has to be invested in manual labour, which allows for more time devotion to analyse feature extraction methods, such as the line of symmetry which is presented here. In the future further analysis is required to see how accurate the segmentation objectively is. This can be done by constructing a dataset with ground truth annotation.

Secondly, phenotyping features (image moments, area, histogram and sum of intensities) are parallelised in order to achieve fast processing. The data-parallelism methodology results in speed ups of 5 to 45 times for the image moments, area, histogram and sum of intensities calculations re-

		Image moments		Area		Histogram		Sum of intensities	
workgroups	localsize	Time (s)	Speedup	Time (s)	Speedup	Time (s)	Speedup	Time (s)	Speedup
5120	1	49,83	2,74	14,89	2,23	106,12	0,07	134,84	1,25
2560	2	27,14	5,03	8,7	3,81	57,87	0,12	74,13	2,27
640	8	9,1	15,01	6,06	5,47	23,45	0,31	22,78	7,38
160	32	4,62	29,57	3,28	10,11	17,64	0,41	12,27	13,70
80	64	3,14	43,50	3,5	9,47	6,54	1,10	9,67	17,39
20	256	3,02	45,23	2,46	13,48	4,66	1,54	9,48	17,74

Table 1. The results of the calculations of the 4 features on 5120 images (with a resolution of 300x300 pixels) done a 100 times on the GPU. The time is given is seconds, the speedup indicates how much faster the implementation is compared to a single threaded implementation on the CPU.

	Image moments		Area		Histogram		Sum of intensities	
Threads	Time (s)	Speedup	Time (s)	Speedup	Time (s)	Speedup	Time (s)	Speedup
1	136,6	1	33,16	1	7,17	1	168,14	1
2	72,11	3,16	18,1	1,83	3,18	2,25	90,16	1,86
4	43,29	4,55	12,06	2,73	2,19	3,27	64,5	2,61
8	29,99	4,73	9,57	3,46	1,58	4,54	55,29	3,04
16	28,9	4,78	9,47	3,50	1,43	5,01	51,08	3,29
32	31	4,41	9,57	3,46	1,52	4,72	54,38	3,09
64	29,64	4,61	9,49	3,49	1,38	5,20	52,67	3,19

Table 2. The results of the calculations of the 4 features on 5120 images (with a resolution of 300x300 pixels) done a 100 times on the CPU. The time is given is seconds, the speedup indicates how much faster the implementation is compared to a single threaded implementation on the CPU.

spectively. This fast processing on a high end GPU and CPU indicates that when using a less powerful CPU or GPU (e.g. a device with a low price) for industrial plant phenotyping, the images can still be processed very fast. Because of this fast processing there is room for additional features to be added. Apart from industrial analysis, it is also possible to process terabytes of images captured in the past in a very short time span.

Acknowledgement

Special thanks to Michael D'hont and Pieter Reuse for helping with the implementation. The authors also wish to thank Laury Chaerle for setup, image acquisition and annotation of the time lapse sequences.

7. REFERENCES

[1] S Dhondt, D Van Haerenborgh, C Van Cauwenbergh, R. M. H. Merks, W. Philips, G. T. S. Beemster, and D. Inze, "Quantitative analysis of venation patterns of arabidopsis leaves by supervised image analysis," *Plant Journal*, vol. 69, no. 3, pp. 553–563, 2012.

[2] C A. Price, O Symonova, Y Mileyko, T Hilley, and J S. Weitz, "Leaf extraction and analysis framework graphical user interface: Segmenting and analyzing the structure of leaf veins and areoles," *Plant Physiology*, vol. 155, no. 1, pp. 236–245, 2011.

[3] Leister D, Varotto C, Pesaresi P, Niwergall A, and Salamini F, "Large-scale evaluation of plant growth in arabidopsis thaliana by non-invasive image analysis," *Plant Physiology and Biochemistry*, vol. 37, no. 9, 1999.

[4] M Jansen, F Gilmer, B Biskup, K A Nagel, U Rascher, A Fischbach, S Briem, G Dreissen, S Tittmann, S Braun, and et al., "Simultaneous phenotyping of leaf growth and chlorophyll fluorescence via growscreen fluoro allows detection of stress tolerance in arabidopsis thaliana and other rosette plants," *Functional Plant Biology*, vol. 36, no. 11, pp. 902–914, 2009.

[5] S Arvidsson, P Prez-Rodrguez, and B Mueller-Roeber, "A growth phenotyping pipeline for arabidopsis thaliana integrating image analysis and rosette area modeling for robust quantification of genotype effects," *New Phytologist*, vol. 191, no. 3, pp. 895–907, 2011.

[6] A Walter, H Scharr, F Gilmer, R Zierer, K Nagel, M Ernst, A Wiese, O Virnich, M Christ, and B Uhlig, "Dynamics of seedling growth acclimation towards altered light conditions can be quantified via growscreen: a setup and procedure designed for rapid optical phenotyping of different plant species.," *New Phytologist*, vol. 174, no. 2, pp. 447–455, 2007.

[7] J De Vylder, F Vandenbussche, Y Hu, W Philips, and D Van Der Straeten, "Rosette tracker: An open source image analysis tool for automatic quantification of genotype effects," *Plant Physiology*, vol. 160, no. 3, pp. 1149–1159, 2012.

[8] Eric W, "Moore neighborhood from math world-a wolfram web resource," `http://mathworld.wolfram.com/MooreNeighborhood.html`.