3-D Histogram-Based Segmentation and Leaf Detection for Rosette Plants

Jean-Michel Pape[1] and Christian Klukas[1]

Paper ID 28

Abstract. Recognition and segmentation of plant organs like leaves is one of the challenges in digital plant phenotyping. Here we present a 3-D histogram-based segmentation and recognition approach for top view images of rosette plants such as *Arabidopsis thaliana* and tobacco. Furthermore a Euclidean-distance-map-based method for the detection of leaves and the corresponding plant leaf segmentation was developed. An approach for the detection of optimal leaf split points for the separation of overlapping leaf segments was created. We tested and tuned our algorithms for the Leaf Segmentation Challenge (LSC). The results demonstrate that our method is robust and handles demanding imaging situations and different species with high accuracy.

Keywords: 3-D Histogram Thresholding, Distance Map, Graph Analysis, Leaf Counting, Leaf Segmentation

1 Introduction

The analysis of digital plant images is an important task in phenotyping to evaluate plant parameters in a non-invasive fashion. A wide variety of different screening systems with varying requirements to the image analysis have been developed and are in part commercially available. Fully automated systems try to establish constant environments for image acquisition, but due to the high costs, space requirements and installation effort of those systems the utilization of more flexible ad-hoc installations would often be desirable. The demanding non-constant imaging situations with respect to varying plant background and fluctuating illumination cause similar problems for image analysis as field-based imaging. Challenging are big differences in image-quality like image resolution and lightning conditions, which need to be handled by image-processing algorithms. Improvements in these areas, would allow an easier monitoring of plant growth in non-automated greenhouses and would also be useful for improved imaging-based field-phenotyping solutions.

[1] Department of Molecular Genetics, Leibniz Institute of Plant Genetics and Crop Plant Research (IPK), Corrensstrasse 3, D-06466 Gatersleben, Germany.

State of the Art Software A comprehensive overview about phenotyping software can be found at http://www.plant-image-analysis.org/. There are a various number of applications which support fully automated or semi-automated plant image analysis, especially for rosette plants, as described in [1],[2],[3],[4],[5] and [6]. Some tools already provide general pipelines for shoot analysis and different plant species including the possibility for rosette plant analysis [7]. In most biological experiments which are designed to be analyzed by automated imaging solutions the growth conditions are modified in comparison to normal field and greenhouse growth, and pot handling conditions. For example, instead of soil, nutrient solutions are used for root phenotyping, and special carrier systems and pot soil covering solutions are used in automated greenhouses. One of the goals of these modifications is to ensure that in respect to the imaging conditions the input data is as homogeneous as possible. However, to reduce effort and cost for setting up high-throughput phenotyping experiments, it is desirable to handle even disturbed images by image analysis tools. Image analysis frameworks such as ImageJ and Fiji include state of the art image processing algorithms which can be utilized for algorithm and framework development [8], [9]. To enhance the robustness of segmentation approaches texture features can be utilized [10], additionally active contours are used to improve segmentation [11]. Active Contours are also used for leaf shape classification [12]. Nevertheless, including these algorithms and methods in a framework which is applicable for high-throughput analysis proves to be challenging due to the storage and processing requirements and the need for processing plant identifiers and meta-data.

2 Methods

The main steps of our method are depicted in figure 1. After image acquisition the pre-processing procedures are performed. Based on the training data two 3-D color-histograms for foreground and background data are calculated, they are used in the segmentation phase to separate the testing image set into foreground and background. The segmentation results are further processed in the feature extraction phase to detect the leaf segments. This involves the detection of leaf center points and skeleton generation. Skeleton-points with minimal distance to the background are starting points for the calculation of split lines. These lines are used as borders during segmentation of overlapping leaves. In a last step the separated leaves are labeled by a region-growing algorithm. Our methods development are related to a dataset provided through the Leaf Segmentation Challenge (LSC) of the Computer Vision Problems in Plant Phenotyping (CVPPP 2014) workshop organized in conjunction with the 13th European Conference on Computer Vision (ECCV) [13]. The dataset is used for testing the methods, further details are provided in the results section.

2.1 Image Acquisition

Our segmentation approach requires plant images and manually labeled images as input for the training phase. Within the Leaf Segmentation Challenge (LSC)

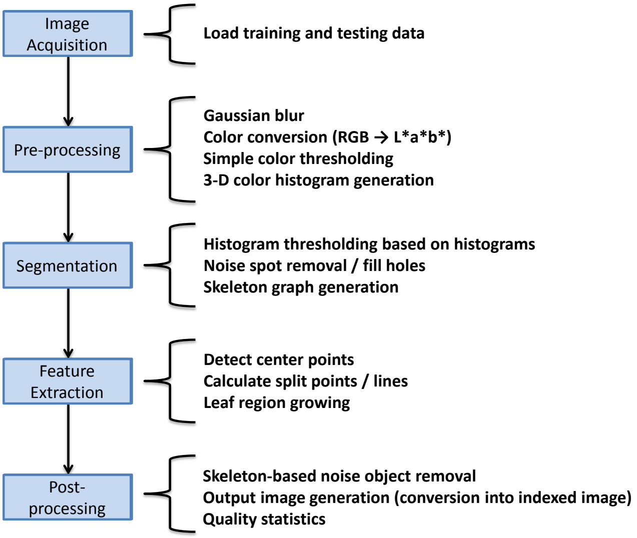
Image Acquisition
Load training and testing data
Pre-processing
Gaussian blur
Color conversion (RGB → L*a*b*)
Simple color thresholding
3-D color histogram generation
Segmentation
Histogram thresholding based on histograms
Noise spot removal / fill holes
Skeleton graph generation
Feature Extraction
Detect center points
Calculate split points / lines
Leaf region growing
Post-processing
Skeleton-based noise object removal
Output image generation (conversion into indexed image)
Quality statistics

Fig. 1: Method overview, main pipeline steps based on the traditional image processing pipeline.

a comprehensive set of images and label data (so called ground-truth data) has been made available. There are three training datasets: Two *Arabidopsis thaliana* plant image datasets with 95 (A1) and 31 (A2) images, and one dataset consisting of 27 tobacco plant images (A3) (fig. 2). The datasets A1 and A2 are similar with respect to their image quality (resolution A1: 500×530 px, A2: 530×565 px). A1 includes more artifacts, e.g. moss. The background and lightning conditions are homogeneous. In opposite, the dataset A3 has a much better image quality (resolution 2448×2048 px), but the background and lightning conditions are very in-homogeneous, also the plant is not strictly located in the image center and other plants are partially visible at the image borders, parts of some of the plants are cut off at the image borders.

2.2 Preprocessing

*L*a*b* Color Space Conversion* All RGB images are converted into the L*a*b* color space (color components are normalized and discretized between 0 - 255). Using L*a*b* channels as features for segmentation has some advantages over using the RGB color space. In comparison to the RGB color space the L*a*b* color components are better suited to separate foreground and background, also the color components are less correlated to each other [14].

Simple Color Thresholding To prevent influences of very dark and very bright pixels to the training data, a color thresholding is applied. These pixels with a

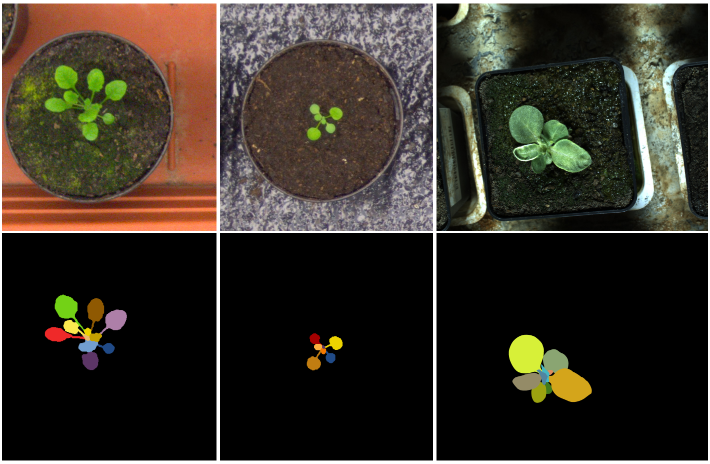

Fig. 2: Example training images from datasets A1, A2 and A3 (left, middle, right). Top - RGB images, bottom - provided ground-truth label images, representing desired optimal thresholding and leaf segmentation results.

L-value near the white and black point are mostly the result of an overexposure, reflections or shadows and include no reasonable color information.

Creation of Color Cubes The segmentation approach based on a supervised classification in foreground and background orientated on the kernel density estimation approach. Therefore a 3-D histogram creation for all training images (with labels) from a given dataset A1, A2 and A3 are processed individually. Each pixel from the training image is categorized into foreground or background by inspecting the provided label data. The corresponding L*a*b* pixel color values are used as indices for the 3-D histogram cubes. For each pixel the corresponding histogram bin is incremented. During this procedure a overall foreground and background 3-D histogram is accumulated. To improve the robustness of the thresholding approach, all input images for the cube calculation were filtered in the pre-processing phase by a Gaussian blur operation.

2.3 Segmentation

As described in Kurugollu et al. a simple histogram thresholding for each channel would result in a partitioning of the 3-D histogram into rectangular regions [15] with non-optimal results. For this reason a direct look-up in the 3-D histogram cubes instead of (multiple) one-dimensional color component thresholds are used: The cubes act as a look-up table which stored the classification probabilities for each color feature. The indices for look-up of the histogram values belonging to particular pixel color, are again based on the discretized L*a*b* color values. For color values not present in the training data, the histogram values are zero. In such a case the surrounding of the particular histogram cell is considered by calculating down-sampled cubes, containing the average of multiple adjacent cells. The histogram values are then interpreted as probabilities and the pixels are therefore assigned to foreground if the corresponding cube contains a higher value than the cell of the background cube.

As the color information is not sufficient to separate the image error-free, the results still include noise and artifacts. As shown in figure 3, it becomes obvious that a simple multiple histogram thresholding would not result in a good segmentation quality, especially the foreground and background components in the A3 dataset contain many overlapping areas.

To handle this disturbances a connected components detection is performed to delete artifacts with an area below a certain threshold. Background areas within the filled image area are also investigated according to their size, and filled, if they fall below a threshold. Morphological operations are used to smooth the object borders. In case of the A3 images, plants are not strictly located at the center of the image and other plant parts protrude into the image from the side. Therefore, all foreground parts which are connected to the border are removed (e.g. leaves from neighbor plants), except if this removal operation would remove the largest connected component.

Remaining large greenish objects within the image are further evaluated in the post-processing phase, once structural shape information (needed for the leaf segmentation), is available.

2.4 Feature Extraction

The segmentation results serve as input for the leaf detection. Especially the leaves of the *Arabidopsis thaliana* plants are considered as compact objects which only partly overlap. In the corresponding euclidean distance map (figure 4 top right) the leaf center points appear as peaks. Before calculating the distance map a morphological erode operation is performed for a better separation of leaves. The Euclidean distance map (Edm) is processed by a maximum search. The result is shown in the bottom left of figure 4. Slightly overlapping leaves are in still detected separately. In cases where overlapping leaves form a single compact object this approach may fail to detect specific leaves. Finally, a skeleton image is calculated for the subsequent analysis steps.

Graph Representation The plant leaves are mostly connected with each other (either overlapping or connected by the plant center). To detect split points for leaf-separation, a graph structure for efficient traversal of the plant mask image skeleton is generated (see fig. 5). Before generating the graph, values of the calculated distance map are mapped on the skeleton image. The result image is used for creation of the skeleton graph: Leaf center points, skeleton end-points and skeleton branch-points are represented as nodes in the graph. Edges are created if the according image points are connected by the skeleton. Additionally, a list of the positions and minimal distances of each particular edge segment is saved as an edge-attribute. This list is used to detect the exact positions of the leaf split points.

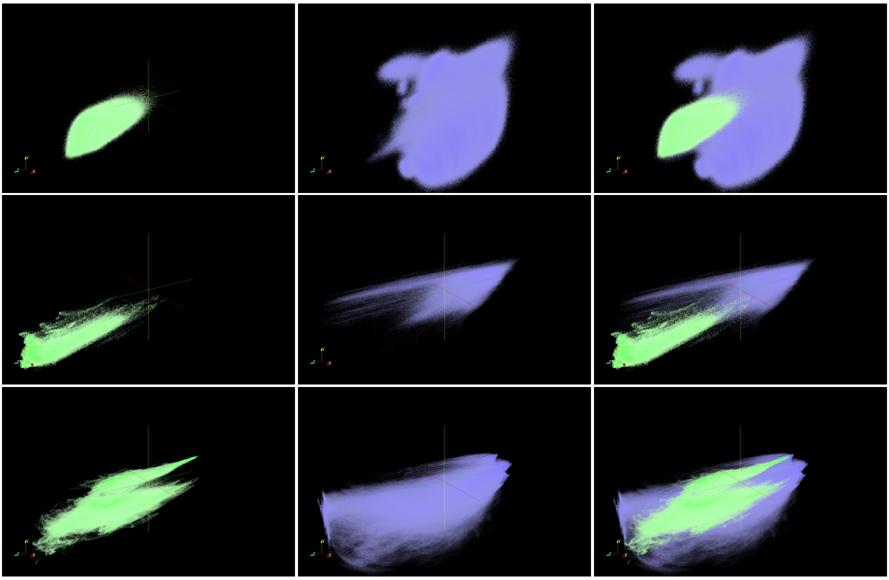

Fig. 3: Accumulated foreground (green) and background (blue) probabilities, stored in 3-D histogram cubes derived from all images of the three training datasets (A1 in first line, A2 second line, A3 third line). For illustration the cube cell values were normalized between 0 - 255 and converted to 8-bit grayscale TIFF images and then visualised using ParaView [16]. Afterwards the values were mapped to green (foreground) and blue (background) color table (left and middle of the image, combined view in the third column). Light colors indicates low values (and thus a low probability) and saturated colors indicate high values. L*a*b* color axes: z-axis: L-value, x-axis: a-value, y-axis: b-value.

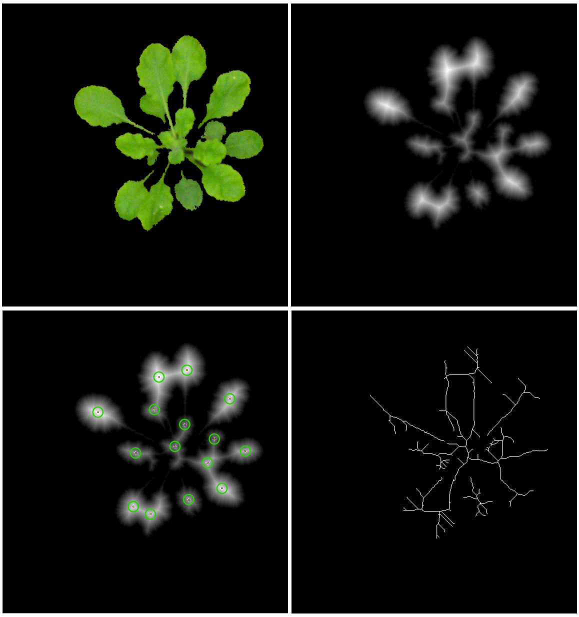

Fig. 4: Segmentation result (top left), distance map (top right), distance map with highlighted peaks, which serve as leaf center points (bottom left) and skeleton image (bottom right).

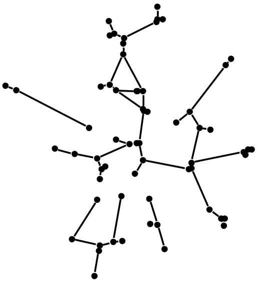

Fig. 5: Derived graph from leaf center points and skeleton image.

Split Point and Split Lines Estimation To separate all leaves from each other, all paths between the leaves are investigated using the corresponding graph structure. The minimum distance points (points where the distance to the image background is minimal) between any two leaf center point nodes are determined by investigating the path edges minimum distance attributes and saved as leaf split points. The according edges are removed from the graph structure. This procedure continues until all leaf center point nodes in the graph are disconnected from each other. Based on the calculated split points the exact split lines are needed to separate overlapping leaves (see fig. 6). For each split point the nearest background point is searched. The second coordinate of the split line is searched at the opposite position relative to the split point (a background pixel near the opposite point but with minimum distance to the split point). After the split line estimation a region filling, considering the segmentation result and the split line positions is performed starting from the leaves center points. The result represents the leaf labels.

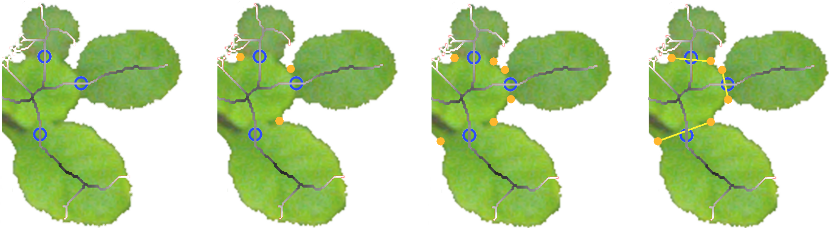

Fig. 6: Example for split point and split line estimation. For illustration the euclidean distance map derived skeleton is mapped on the segmented plant image (gray values indicate the euclidean distance to the background). (from left to right) Identified split points, detected start points for split line detection (nearest outline points to the individually split point), corresponding endpoints for split lines, resulting split lines.

2.5 Post-processing

During the segmentation phase only color and size information is considered for artifact removal. For the A3 dataset and the large greenish noise objects, the structural information from the skeleton and graph structure is evaluated. The average distance from node to node is calculated for each connected component. While the shape of plant objects is relatively compact the noise objects contain many skeleton branch points. Therefore, the average distance for noise objects is small. To increase the difference of this property for plant and noise objects the distance is scaled according to the average distance of the object relative to the image center. Objects at the image border are then more likely removed.

The last step of the workflow includes the output image generation (labeled result images) and the measurement of quality statistics based on the provided evaluation functions.

3 Implementation

Our approach is implemented in Java, taking advantage of its platform independence and the availability of numerous libraries like ImageJ and Fiji. As shown in figure 7, the pipeline consists of four main blocks. The provided training images and their labels are used to calculate the foreground and the background 3-D color histogram cubes. These cubes are then used in a first segmentation phase to process the provided testing images and extract the foreground and background. The segmentation result is used to detect leaves by detecting leaf center points and the corresponding split points and split lines based on distance map and skeleton calculation. In the last step the region growing algorithm labels each leaf region.

Pipeline Parameters Besides the trained 3-D histogram cubes several parameters influence the segmentation and leaf detection. Individually for the three datasets well suited parameter values were selected. Depending on the dataset noise level in the pre-processing according blurring factors, noise removal and gap fill size limits for disconnected components were determined. The segmentation results were further improved by introduction of a weighting factor in order to increase the probability for detection of foreground pixels. This way the plant is better recognized, additionally introduced noise objects are removed if they fall below the noise area limit or during the post-processing based on their irregular shape.

4 Results and Discussion

Training Results The images (fig. 8 and fig. 9) show the result of different pipeline-steps. Table 1 contains the statistical results of the leaf area labeling (column 1), foreground/background separation (column 2) and leaf detection (average absolute and mean errors per image in column 3 and 4) of the training data. The foreground and background separation of the three datasets is nearly optimal (97.4 - 99.7%).

Testing Results The result for the testing data are shown in table 2. The foreground and background segmentation and the leaf labeling was performed mostly successfully with similar results as for the training data (fig. 10).

For the testing phase three datasets, belonging to the training data without the according ground-truth images have been provided by the organizers of the Leaf Segmentation Challenge. The test data images for A1 (33) and A2 (9) are very similar to the training data, the 56 A3 test images show more differences to the training data in respect to the imaging background and plant colorization.

Overall, the results of the test data are similar to those of the training data. Problematic for segmentation was discoloration of some of the images in the A3 dataset. In one case the whole (very small plant) was removed completely, as the cut-off value for the size of noise objects was tuned for the smallest plants in the training dataset and proved to be too high for the testing-data. The quality of

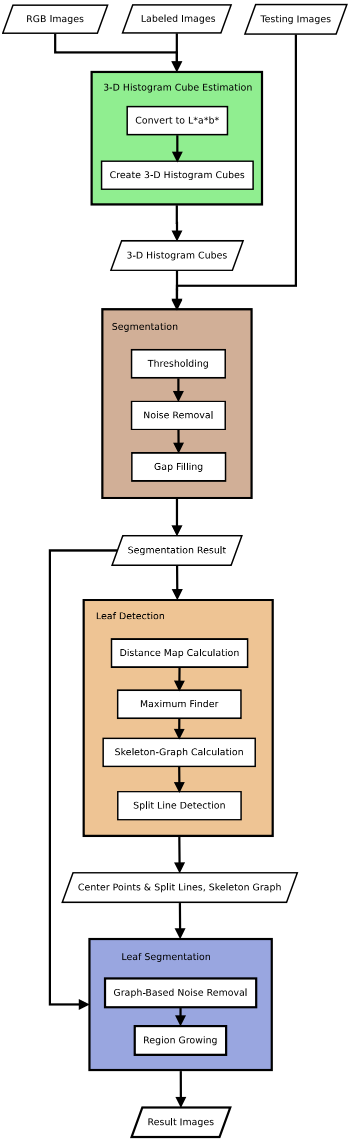

Fig. 7: Design of the implemented processing pipeline. Green: training phase including the histogram estimation for foreground and background. Brown: segmentation and noise removal. Orange and Blue: Extraction of features for

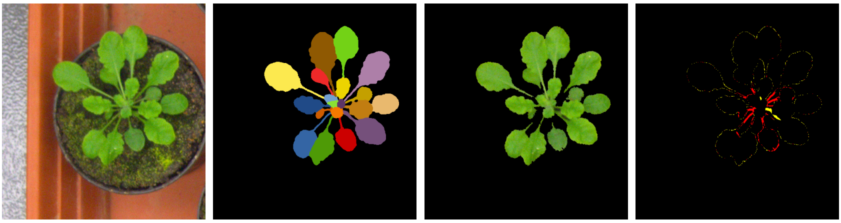

Fig. 8: From left to right: input image, provided image label, segmentation result, color coded difference image (yellow - false positive, red - false negative).

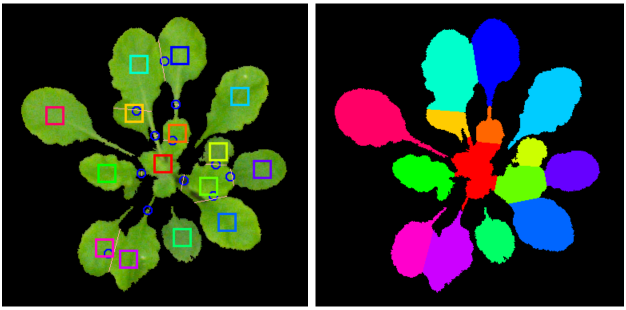

Fig. 9: Left: Leaf center points (rectangles), split points (blue circles), split lines (orange lines). Right: Result of the leaf segmentation.

Table 1: Results of the evaluation of the training data. BestDice: Quality of the idividual leaf segmentation. FGBGDice: Quality of the foreground and background separation. AbsDiffFGLabels: Average absolute difference of the number of the detected leaves. DiffFGLabels: Average difference of the detected number of leaves. For all values the standard derivation is indicated. Calculation details are described in [13].

	BestDice [%]	**FGBGDice [%]**	**AbsDiffFGLabels**	**DiffFGLabels**
A1	74.2 (±7.7)	97.4 (±1.8)	2.6 (±1.8)	-1.9 (±2.5)
A2	80.6 (±8.7)	99.7 (±0.3)	0.9 (±1.0)	-0.3 (±1.3)
A3	61.8 (±19.1)	98.2 (±1.1)	2.1 (±1.7)	-2.1 (±1.7)
all	73.5 (±11.5)	98.0 (±1.9)	2.2 (±1.7)	-1.7 (±2.3)

Table 2: Statistical evaluation results provided by the Leaf Segmentation Challenge board, based on the submitted image analysis results for the testing-dataset.

	BestDice [%]	**FGBGDice [%]**	**AbsDiffFGLabels**	**DiffFGLabels**
A1	74.4 (±4.3)	97.0 (±0.8)	2.2 (±1.3)	-1.8 (±1.8)
A2	76.9 (±7.6)	96.3 (±1.7)	1.2 (±1.3)	-1.0 (±1.5)
A3	53.3 (±20.2)	94.1 (±13.3)	2.8 (±2.5)	-2.0 (±3.2)
all	62.6 (±19.0)	95.3 (±10.1)	2.4 (±2.1)	-1.9 (±2.7)

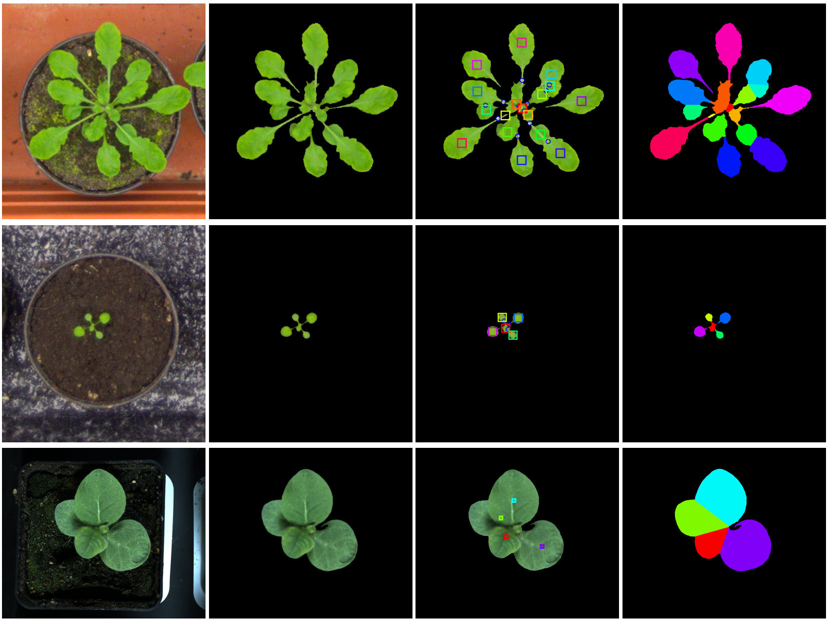

Fig. 10: Example results from the evaluation-phase. From left to right: input image, segmentation result, detected leaf center points, labeled leaves. Test-images from top to bottom: plant 87 from dataset A1, plant 10 from set A2 and plant 47 from set A3.

our segmentation approach depends on the homogeneity of the training data in comparison to the testing data. The training dataset needs to be representative, it would desirable to improve the interpolation of missing points (determination of probabilities for unknown color values) in the histogram cubes. The current scale space method is an inaccurate approximation, a better option would have been a blurring operation in the 3-D space.

In the segmentation example for A1 it is noticeable that the petioles are missing in some images for some leaves. An explanation could be that the color of these thin areas is similar to the moss and other background parts in the A1 dataset.

A remaining challenge is the recognition of very small leaves and leaves which overlap strongly. Figure 11 shows some examples for the leaf center point detection based on the euclidian-distance-map and maxima detection. In addition, the leaf segmentation could perform better at leaf borders which overlap. Due to current implementation issues regarding the discretization of the distance map, the split lines sometimes don't directly connect points of minimal distance. In addition, within our approach, it is not clear which leaf overlaps the other and therefore a straight line is constructed for separation. By analyzing the leaf area next to the line and the borders of the leaves in the surrounding, a better fitting curve could be estimated. The average leaf count results are too low for all three datasets (DiffFGLabel-values of -1.8 for A1, -1.0 for A2 and -2.0 for A3). This is

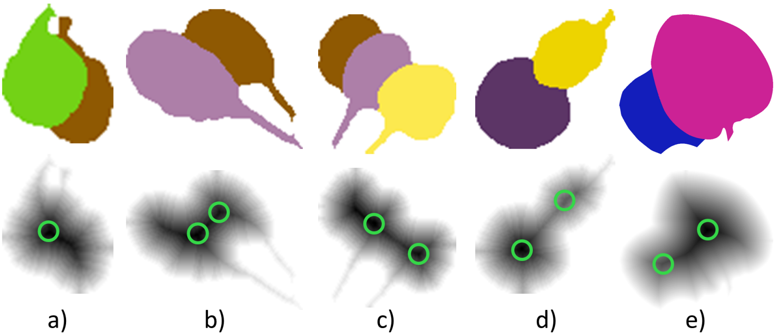

Fig. 11: Examples for overlapping leaves and the corresponding euclidean-distance-map with detected maxima. a) - d) Examples for arabidopsis, e) example for tobacco. Correctly identified are b), d) and e), too few maxima are observed in case a) and c). The distance map is not fine granular enough in these cases.

mainly caused by very small leaves which are located at the center of the plant, these leaves are not detected as they don't appear as peaks in the distance map. The leaf segmentation for the tobacco images of the A3 dataset performs comparably worst (BestDice values of 53% in A3 versus 74 and 77% for A1 and A2). The leaves of the tobacco plants have a different shape than the *Arabidopsis thaliana* plants. In later development stages the leaf overlap becomes so large that our detection of peaks in the distance-maps fails to recognize those plant structures.

5 Conclusions

The leaf separation approach was developed for compact leaf shapes as found in *Arabidopsis thaliana*. Leaves of tobacco plants are not as well separated from each other, while the developed approach still works for tobacco plants.

The calculation of color cubes using the L*a*b* color space proved to be a very good basis for foreground/background separation of images which are not too different from the training data. We also developed a way for the detection of leaf center points using a distance map, and an approach for separation of leaf segments, by calculating split lines.

It is conceivable to use this approach in the future within a semi-automated segmentation method, outside of this specific Leaf Segmentation Challenge. The representative training data could be created by the user by marking image regions belonging to foreground and background. In addition, the leaf segmentation approach could be improved by a an shape-adjusting component.

References

1. Tessmer, O.L., Jiao, Y., Cruz, J.A., Kramer, D.M., Chen, J.: Functional approach to high-throughput plant growth analysis. BMC systems biology **7**(Suppl 6) (2013) S17
2. Walter, A., Scharr, H., Gilmer, F., Zierer, R., Nagel, K.A., Ernst, M., Wiese, A., Virnich, O., Christ, M.M., Uhlig, B., et al.: 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 **174**(2) (2007) 447–455
3. Ispiryan, R., Grigoriev, I., zu Castell, W., Schffner, A.R.: A segmentation procedure using colour features applied to images of arabidopsis thaliana. Functional Plant Biology **40** (2013) 1065–1075
4. Bours, R., Muthuraman, M., Bouwmeester, H., van der Krol, A.: Oscillator: A system for analysis of diurnal leaf growth using infrared photography combined with wavelet transformation. Plant methods **8**(1) (2012) 29
5. Green, J.M., Appel, H., Rehrig, E.M., Harnsomburana, J., Chang, J.F., Balint-Kurti, P., Shyu, C.R.: Phenophyte: a flexible affordable method to quantify 2d phenotypes from imagery. Plant methods **8**(1) (2012) 45
6. De Vylder, J., Vandenbussche, F., Hu, Y., Philips, W., Van Der Straeten, D.: Rosette tracker: an open source image analysis tool for automatic quantification of genotype effects. Plant physiology **160**(3) (2012) 1149–1159
7. Klukas, C., Chen, D., Pape, J.M.: Integrated analysis platform: An open-source information system for high-throughput plant phenotyping. Plant physiology **165**(2) (2014) 506–518
8. Schneider, C.A., Rasband, W.S., Eliceiri, K.W., Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., Preibisch, S., et al.: 671 nih image to imagej: 25 years of image analysis. Nature methods **9**(7) (2012)
9. Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., Preibisch, S., Rueden, C., Saalfeld, S., Schmid, B., et al.: Fiji: an open-source platform for biological-image analysis. Nature methods **9**(7) (2012) 676–682
10. Valliammal, N., Geethalakshmi, S.: Leaf image segmentation based on the combination of wavelet transform and k means clustering. International Journal of Advanced Research in Artificial Intelligence **1**(3) (2012) 37–43
11. Minervini, M., Abdelsamea, M.M., Tsaftaris, S.A.: Image-based plant phenotyping with incremental learning and active contours. Ecological Informatics (2013)
12. Cerutti, G., Tougne, L., Vacavant, A., Coquin, D.: A parametric active polygon for leaf segmentation and shape estimation. In: Advances in Visual Computing. Springer (2011) 202–213
13. Scharr, H., Minervini, M., Fischbach, A., Tsaftaris, S.A.: Annotated image datasets of rosette plants. Technical report (2014)
14. Bansal, S., Aggarwal, D.: Color image segmentation using cielab color space using ant colony optimization. International Journal of Computer Applications **29**(9) (2011) 28–34
15. Kurugollu, F., Sankur, B., Harmanci, A.E.: Color image segmentation using histogram multithresholding and fusion. Image and vision computing **19**(13) (2001) 915–928
16. Henderson, A., Ahrens, J., Law, C.: The ParaView Guide. Kitware Clifton Park, NY (2004)