MUSEED: A MOBILE IMAGE ANALYSIS APPLICATION FOR PLANT SEED MORPHOMETRY

Ke Gao[1], Tommi White[2], Kannappan Palaniappan[1], Michele Warmund[3], Filiz Bunyak[1]

[1]Department of Electrical Engineering and Computer Science
[2]Department of Biochemistry
[3]Division of Plant Sciences
University of Missouri-Columbia, MO, USA

ABSTRACT

This paper presents an Android-based mobile application named MUSeed that automatically computes seed morphometry utilizing image processing techniques. Unlike most of the existing tools, MUSeed does not impose restrictions on arrangement of seeds since it is capable of handling touching seed instances. First, RGB color space is converted to RG-chromaticity space to reduce the influence of shadows and illumination variations. Then K-means clustering is performed to segment the seeds from the background. To split touching seeds in the image, a Watershed with Edge-Augmented Markers (WEAM) algorithm and Concave Point Analysis (CPA) are developed. After that, a fitness function is performed to select the most appropriate result. MUSeed is benchmarked against other similar tools on 7 cultivars of American elderberry seeds and shows promising results.

Index Terms— plant seed morphometry, image analysis, mobile application

1. INTRODUCTION

A seed sustains the beginning stage of a plant [1]. Its formation is a vital aspect of germination survival and subsequent development of a plant's new generation [2] [3]. Some seeds such as rice and maize are the primary agricultural crops for human consumption. They are serving as the major types of staple food for a large percentage of people in the world. Studies indicate that seed/grain kernel size and shape are crucial since they represent one of the major components of yield [4]. So it is important to develop an accurate, low-cost, and high-throughput morphometry method that allows for a detailed analysis of seed samples for the purpose of testing plant yield and vigor.

Several tools based on image analysis have been developed for seed morphometry. Some are integrated systems that consist of both dedicated hardware and software components. Systems like SeedCount [5] depend on an optical scanner to produce high resolution images free of illumination variations. They utilize specifically designed software to generate detailed measurements for seed size and shape. However, it requires a considerable length of time for sample preparation as the seeds must be carefully placed either on a tray or a scanner. Ensuring that the seeds are not touching each other is particularly hard for small seeds such as elderberry or Arabidopsis seeds. The initial cost of these systems is also prohibitive.

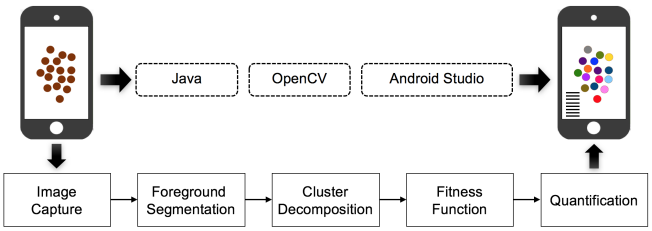


Fig. 1. MUSeed flowchart.

There are some free tools for seed morphometry using image analysis. SmartGrain [6] and SeedSize [1] are dedicated seed morphometry software. Both software requires the seeds to be spread uniformly over the scanner because they cannot deal with touching seeds. ImageJ [7] is a cross platform and open source image processing program. The general watershed algorithm in ImageJ can be used to segment touching seeds. GrainScan [4] is a Microsoft Windows-based software that can separate touching seeds. It requires a fixed background color and can only handle small groups of touching seeds. Reflection and shadow should also be minimized during image capture.

In this paper, we present an Android-based mobile application (MUSeed) with easy image capture and automated measurements (Fig 1). It does not require dedicated hardware. It is capable of separating touching seeds, so no restrictions are imposed on how the seeds are arranged. Section 2 describes the image analysis pipeline and mobile application

Thanks to MU Information Technology Committee and MU Interdisciplinary Innovations Fund for funding the project SmartMScope: A Microscopy Image Analysis App for Mobile Platforms.

development details. Section 3 presents experimental results on seven sample seed cultivars. Section 4 concludes the paper and presents our future directions.

2. METHOD

K-means clustering is used to segment the seeds from the background. Mathematical morphology operations are performed to refine the segmentation. After connected components are labelled, clusters consisting of multiple touching seeds are identified based on size and shape criteria for further processing by the cluster decomposition module. Finally, shape and size parameters are automatically computed for the individual seeds identified by the segmentation and cluster decomposition modules.

2.1. Cluster Decomposition

Accurate segmentation of individual seeds within a cluster is crucial for morphometry operations. While small clusters of 2-3 seeds can be resolved with simple approaches. Decomposition of larger clusters are challenging.

For robust and accurate cluster decomposition, candidate segmentations are generated with complementary region-based (Watershed with Edge-Augmented Markers) and boundary-based (Concave Point Analysis) systems. The final segmentation is determined with a novel automated evaluation and selection scheme that operates on multi-scale and multi-method outputs. The selection process ensures robustness under different seed arrangements where reliable region or boundary information may not be available.

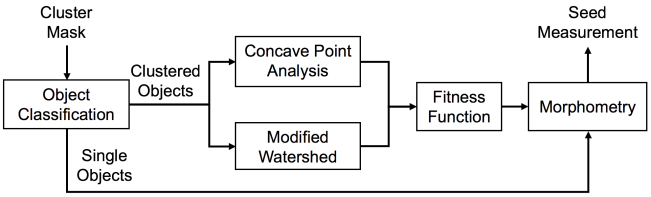
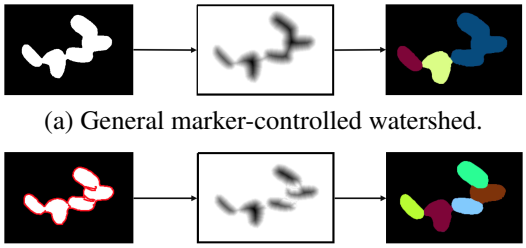


Fig. 2. Cluster decomposition.

2.1.1. *Watershed with Edge-Augmented Markers*

Watershed with Edge-Augmented Markers (WEAM) combines the general marker-controlled watershed segmentation algorithm [8] with Canny edge detection [9]. The general marker-controlled watershed relies on additional sources to mark the rough location of the objects (internal markers) and the background (external markers). To a great extent, segmentation performance is determined by the markers (Fig 3). Therefore, the proposed algorithm includes the Canny edge detection to create more accurate internal markers by locating the area where two objects are touching each other. While combination of Canny edge detection and Watershed segmentation have been previously explored [10] [11] [12]. These studies incorporate edge information to refine object boundaries, but fail to address the problem of incorrect markers that leads to under or over segmentation of the clusters, which has much more drastic effect than boundary precision.



(a) General marker-controlled watershed.

(b) Watershed with edge augmented markers.

Fig. 3. Segmentation performance comparison.

The proposed edge-augmented marker generation method ensures accurate number and location of internal markers by fusing shape information derived from color-based segmentation with gradient/edge information from the original image (Fig 4). It utilizes Canny edge detector to extract boundary of foreground objects (seeds). The slender region where two seeds meet can also be detected because this region is relatively dark. An example is shown in Fig 5 where one of the touching seed shadow is highlighted. Since the Canny edge detection depends on the parameter selection, a fitness function will be used to determine the optimal parameters. Details will be discussed in section 2.1.3.

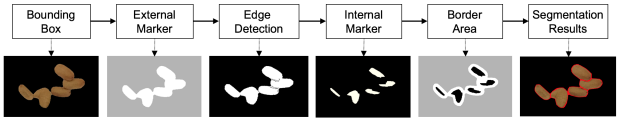


Fig. 4. Modified marker-controlled watershed workflow.

The seeds to be segmented come in all colors and textures. Heterogeneous texture or even dark-colored spots might be mistakenly identified as a touching region. So a Gaussian filter is used to remove noise. A proper standard deviation of Gaussian kernel will also be selected by the fitness function.

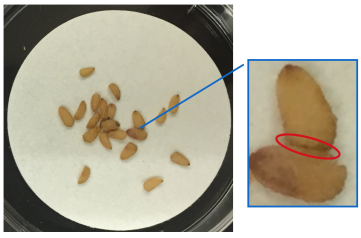


Fig. 5. Zoomed image showing the touching seeds.

The WEAM includes four steps:

Step 1. The external marker indicating the background area is obtained using morphological dilation on the binary mask of an image.

Step 2. Canny edge detection is performed on the original RGB image. The binary mask is overlaid with the detected edges.

Step 3. The Euclidean distance transform is applied to the edge-augmented binary mask. The local maxima on the distance map are selected as the internal markers.

Step 4. Apply marker-controlled watershed algorithm using both original RGB image and markers as inputs.

2.1.2. Concave Points Analysis

For the boundary-based cluster decomposition, we use the Concave Point Analysis (CPA). In [13], concave contact points are detected using a circular sliding kernel. Performance of the detection is affected by the size of the kernel. To avoid false positive detection, we validate the detected concave points by their distance to the closest convex hull point as in [14]. Our CPA workflow is illustrated in Figure 6. It includes four major steps.

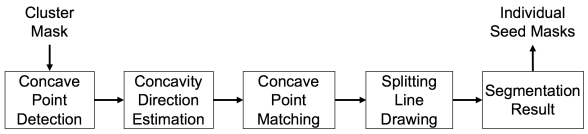


Fig. 6. Concave point analysis workflow.

Step 1. Concave points are detected based on a sliding kernel method [13] and filtered out with distance to convex hull [14].

Step 2. For each concave point, direction of concavity is estimated.

Step 3. Concave points are matched according to a set of criteria including direction of concavity and boundary crossing information as described in [15] [16].

Step 4. Line joining the matching concave points are drawn to split the cluster.

2.1.3. Segmentation Selection using Fitness Function

The process of cluster decomposition depends on several parameters. This module optimizes parameter selections by using a fitness function. As previously discussed, WEAM handles seeds of different shapes and colors by adjusting two parameters: Gaussian kernel standard deviation and lower threshold of Canny edge detector. A fitness function is performed here to evaluate possible combinations of parameters so it can find the best segmentation result. The function is defined as follows:

$$f_1 = |n - n_e| \tag{1}$$

$$f_2 = \frac{|a_e - a_\mu|}{a_\mu} + \frac{|w_e - w_\mu|}{w_\mu} + \frac{|h_e - h_\mu|}{h_\mu} \tag{2}$$

$$f_3 = \sigma_a + \sigma_w + \sigma_h \tag{3}$$

$$fitness = \frac{1}{f_1 + f_2 + f_3} \tag{4}$$

where n is the number of connected components in the segmentation result. n_e is the estimated number of objects. a_e, w_e, and h_e are the estimated area, width, and height of a regular-sized seed, respectively. Likewise, a_μ, w_μ, and h_μ are the average area, width, and height of connected components. σ_a, σ_w, and σ_h are the standard deviation of area, width, and height of connected components. The segmentation result with the maximum fitness value is selected as the final solution.

Fig. 7. (a) Original touching seeds image. (b) CPA result. (c) WEAM result #1. (d) WEAM result #2. (e) WEAM result #3. (f) Selected result.

An example is shown in Figure 7. Multiple touching seeds segmentation results are generated by CPA and WEAM modules. Over-segmentation in Fig 7 (c) and under-segmentation in Fig 7 (e) result in low fitness values. Fitness of Fig 7 (b) and Fig 7 (d) are close, but the latter is slightly higher. So Fig 7 (d) is chosen to be the final segmentation result (Fig 7 (f)).

2.2. Mobile Application Development

Based on the proposed seed segmentation algorithm, an Android-based mobile application (MUSeed) for seed morphometry has been developed. The application is implemented in Java with OpenCV libraries. The integrated development environment is Android Studio. The application is capable of processing seed images and generating measurements of each seed. The measurements can be exported to a text file and sent to an email address provided by the user. Several major graphical user interfaces are shown in Figure 8.

3. EXPERIMENTAL RESULTS

We have tested the proposed MUSeed app on an elderberry seed study involving seeds from seven different cultivars. And

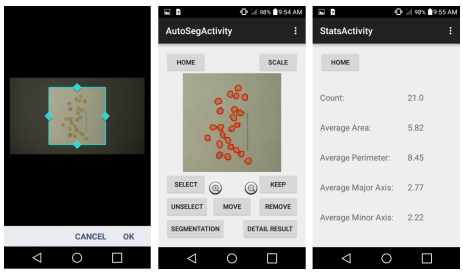


Fig. 8. Major graphical user interfaces of MUSeed

we have compared its performance against GrainScan and ImageJ. As shown in Figure 9, mean and standard deviation of seed area error are computed by subtracting from the manually drawn ground truth. Compared to MUSeed, GrainScan and ImageJ produce larger errors. Figure 10 illustrates a sample set of results. While GrainScan and ImageJ detect 17 and 18 seeds respectively, MUSeed accurately detects all 20 seeds.

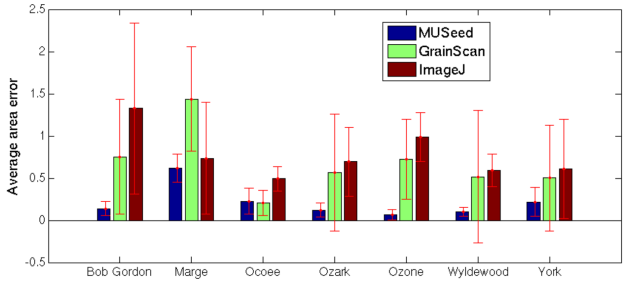


Fig. 9. Mean and standard deviation of seed area error compared to the ground truth

The segmentation performance of three methods are quantified in terms of recall, precision, and F-measure:

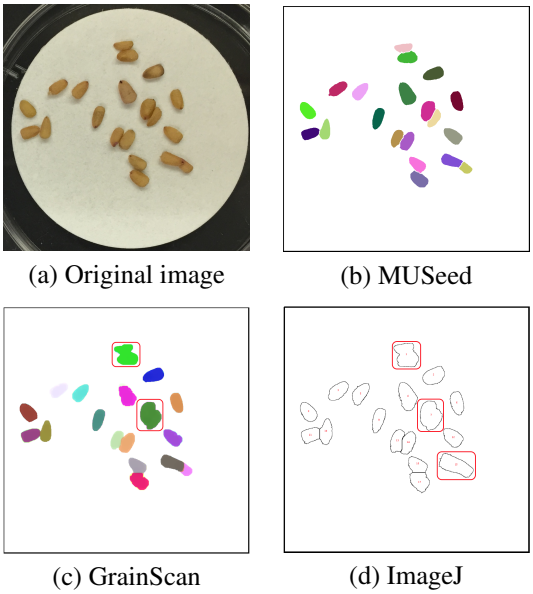
$$recall = \frac{tp}{tp + fn} \tag{5}$$

$$presion = \frac{tp}{tp + fp} \tag{6}$$

$$F = 2 \times \frac{recall \times precision}{recall + precision} \tag{7}$$

where tp, fn, and fp denote the number of true positives, false negatives, and false positives respectively. Each seed in a segmentation image is compared with the ground truth. Those that can be correctly matched are true positives.

Average recall, precision, and F-measure of three methods are computed separately on 7 cultivars (Table 1). The proposed method yields the highest accuracy on each cultivar with an overall average F-measure of 96.66%. The performance of GrainScan and ImageJ are similar in terms of overall average recall, while precisions for both are dramatically lower because of their over-segmentation results and the lack of ability to remove noise especially for ImageJ,



(a) Original image (b) MUSeed

(c) GrainScan (d) ImageJ

Fig. 10. Sample outputs for MUSeed, GrainScan, and ImageJ. Red rectangles mark failed segmentation.

Data	Average recall / precision / F-measure		
	MUSeed	**GrainScan**	**ImageJ**
BG	0.98/0.97/0.97	0.79/0.84/0.81	0.80/0.57/0.67
Mar	0.95/0.98/0.96	0.73/0.53/0.61	0.66/0.21/0.32
Oco	0.95/0.97/0.96	0.90/0.85/0.87	0.92/0.60/0.73
Ozk	1.00/1.00/1.00	0.89/0.90/0.89	0.86/0.51/0.64
Ozn	0.91/0.96/0.93	0.80/0.87/0.83	0.85/0.66/0.74
Wyd	0.97/0.99/0.98	0.81/0.77/0.79	0.83/0.48/0.57
York	0.96/0.94/0.95	0.62/0.52/0.57	0.63/0.32/0.42
Avg	**0.96/0.97/0.96**	0.79/0.75/0.77	0.79/0.48/0.60

Table 1. Average recall, precision, and F-measure of MUSeed, GrainScan, and ImageJ on 7 seed datasets.

4. CONCLUSION AND FUTURE WORK

We have developed an Android-based mobile application (MUSeed) for seed morphometry analysis. The app relies on the built-in camera on a smart device for image capture. It is capable of separating seeds in clusters and measuring the size and shape of each individual seed. To split touching seeds in an image, a Watershed with Edge-Augmented Markers (WEAM) and an improved Concave Point Analysis (CPA) are developed. A fitness function is used to select the most appropriate segmentation result, automatically adapting the app to variations in images due to noise, scale, or seed characteristics. The proposed method is tested on a 7 seed cultivars and evaluated against 2 similar tools. The experiments demonstrate that MUSeed produces measurements of a high accuracy. We are in the process of extending the MUSeed app with additional seed morphometry analysis.

5. REFERENCES

[1] Candace R Moore, David S Gronwall, Nathan D Miller, and Edgar P Spalding, "Mapping quantitative trait loci affecting arabidopsis thaliana seed morphology features extracted computationally from images," *G3: Genes—Genomes— Genetics*, vol. 3, no. 1, pp. 109–118, 2013.

[2] Pam G Krannitz, Lonnie W Aarssen, and Jennifer M Dow, "The effect of genetically based differences in seed size on seedling survival in arabidopsis thaliana (brassicaceae)," *American Journal of Botany*, pp. 446–450, 1991.

[3] Peter Manning, Kelly Houston, and Tess Evans, "Shifts in seed size across experimental nitrogen enrichment and plant density gradients," *Basic and Applied Ecology*, vol. 10, no. 4, pp. 300–308, 2009.

[4] Alex P Whan, Alison B Smith, Colin R Cavanagh, Jean-Philippe F Ral, Lindsay M Shaw, Crispin A Howitt, and Leanne Bischof, "Grainscan: a low cost, fast method for grain size and colour measurements," *Plant methods*, vol. 10, no. 1, pp. 1–10, 2014.

[5] Next Instruments, "SeedCount Digital Imaging of Seeds," `http://www.nextinstruments.net/products/seedcount/`, [Online; Accessed 30-Jan-2017].

[6] Takanari Tanabata, Taeko Shibaya, Kiyosumi Hori, Kaworu Ebana, and Masahiro Yano, "Smartgrain: high-throughput phenotyping software for measuring seed shape through image analysis," *Plant physiology*, vol. 160, no. 4, pp. 1871–1880, 2012.

[7] Caroline A Schneider, Wayne S Rasband, and Kevin W Eliceiri, "Nih image to imagej: 25 years of image analysis," *Nature methods*, vol. 9, no. 7, pp. 671–675, 2012.

[8] Fernand Meyer and Serge Beucher, "Morphological segmentation," *Journal of visual communication and image representation*, vol. 1, no. 1, pp. 21–46, 1990.

[9] John Canny, "A computational approach to edge detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, , no. 6, pp. 679–698, 1986.

[10] Jiayong Yan, Binsheng Zhao, Liang Wang, Andrew Zelenetz, and Lawrence H Schwartz, "Marker-controlled watershed for lymphoma segmentation in sequential ct images," *Medical physics*, vol. 33, no. 7, pp. 2452–2460, 2006.

[11] Wei-Yen Hsu, "Improved watershed transform for tumor segmentation: application to mammogram image compression," *Expert systems with Applications*, vol. 39, no. 4, pp. 3950–3955, 2012.

[12] Lindsay A Baker and John L Rubinstein, "Edged watershed segmentation: A semi-interactive algorithm for segmentation of low-resolution maps from electron cryomicroscopy," *Journal of structural biology*, vol. 176, no. 1, pp. 127–132, 2011.

[13] Qufa Zhong, Ping Zhou, Qingxing Yao, and Kejun Mao, "A novel segmentation algorithm for clustered slender-particles," *Computers and Electronics in Agriculture*, vol. 69, no. 2, pp. 118–127, 2009.

[14] Xue-Ming Qian, Hong Zhu, Chun-Lai Feng, Pei Zhu, Han Li, Wei Xin, and Gang Chen, "An overlapping bubbles partition method in aerated water flows," in *Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference on*. IEEE, 2004, vol. 6, pp. 3746–3750.

[15] P Lin, YM Chen, Y He, and GW Hu, "A novel matching algorithm for splitting touching rice kernels based on contour curvature analysis," *Computers and Electronics in Agriculture*, vol. 109, pp. 124–133, 2014.

[16] Gabriel Fernandez, Murat Kunt, and J-P Zrÿd, "A new plant cell image segmentation algorithm," pp. 229–234, 1995.