 DOI: 10.3217/978-3-85125-954-4-08

An unsupervised, shape-based 3d cell instance segmentation method for plant tissues

Alexander Palmrich[‡], Klara Voggeneder[✎], Danny Tholen[✎], Guillaume Théroux-Rancourt[✎], Jiří Hladůvka[‡], and Walter Kropatsch[‡]
[✎]Institute of Botany, University of Natural Resources and Life Sciences, Vienna
{klara.voggeneder, daniel.tholen, guillaume.theroux-rancourt}@boku.ac.at
[‡]Pattern Recognition and Image Processing Group, Vienna University of Technology
{alexander.palmrich, jiri.hladuvka, walter.kropatsch}@tuwien.ac.at

Abstract

We present a segmentation method for tissue images that uses the shape of image foreground to infer the location of individual cells. The method works in arbitrary dimension and is suited for volumetric scans. It is unsupervised, but allows a user to specify parameters to correct for the presence of noise and to steer the segmentation behavior. After describing the algorithm and its limitations, we analyze its complexity (linear in voxel count) and evaluate the quality of the segmentation result by applying it to a leaf x-ray micro-tomography scan.

keywords: instance segmentation, distance transform, skeleton, watershed, volumetric image, shape, unsupervised

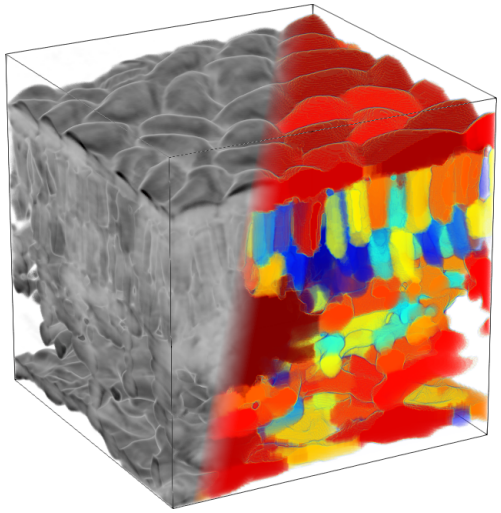


Figure 1. 256^3 voxel X-ray micro-tomography of leaf tissue and its automated cell instance segmentation result

1. Introduction

After obtaining a high resolution 3d image of biological tissue, further processing of that data may require identifying the tissue's constituent cells, i.e. cell instance segmentation. Even if a volumetric scan only allows for differentiation of tissue from background (as in Fig. 1, where the air space in-between cells is considered background), a human expert is still often able to identify single cells. They can infer the position and size of single cells from the shape of the foreground-to-background surface. Due to the large amount of data in a 3d image and the resulting work load, a full segmentation by experts is often not feasible. Our method aims to formalize and automate the human shape-based approach to segmentation. In contrast to other recent methods [8, 14] we do not rely on a neural network that needs training ("*supervised*"), but employ classical geometrical methods from pattern recognition ("*unsupervised*").

Throughout the article, we will use 2d images to illustrate the concepts involved. The actual method operates in arbitrary dimension however, not just on slices of 2d images.

2. Limitations

Our method has the following requirements:

- The image must be an n-d grid (e.g 3d voxels) of grey values.
- The image must fit into RAM.
- The size of the smallest isolated cells to be captured must be known a priori or estimated at run time, to not discard such cells during noise handling.
- Cell shapes must be extractible after a suitable thresholding operation.

- Cell-to-cell interfaces must be smaller in diameter than cell bodies.

3. Algorithm

Our segmentation method can be summarized as following:

1. Extract foreground and background via thresholding.
2. Clean both from noise via morpholigical filtering.
3. Compute distance transform of foreground from background.
4. Assign labels to local maxima of the (smoothed) distance transform.
5. Use watershed to grow the labels.
6. Merge labels, if their border is nearly as wide as their thickest part.
7. Discard labels that are too small.
8. Grow remaining labels again using watershed.

We will now discuss these individual points in more detail.

The distance transform is a great tool to capture shape information, but it is highly sensitive to noise [2]: Even tiny spots of background located in the foreground can severely distort the transform, rendering it useless. This is why careful handling of noise and image artifacts is required. Rather than blurring the source image to reduce noise effects and also discarding potentially important high-frequency data (see Figure 2), we deal with the noise morphologically: After thresholding, we search for connected components of background and *delete* (i.e. assign as foreground) those, whose voxel count is below some specified value. This is a parameter we must specify, and why we must know in advance the size of the smallest structures we wish to capture.

The seed labels for watershed are constructed from both the local maxima of the distance transform (possibly a smoothed distance transform to merge close maxima), and the connected components of the background.

To use watershed, we need a height map that steers the growth of seed labels. Classical watershed segmentation [3] uses the image gradient magnitude as height, which allows for regions to grow fast where the source image has uniform brightness. This however ignores the shape information we have already extracted via the distance transform, and as such is prone to growing labels along image artifacts and noise, which strongly affect the gradient. Instead we choose to use a convex combination of image gradient magnitude and negative distance transform for height value: This allows for region growth to happen from cell centers outwards, uniformly approaching the background, independently from cell size. All the while the image gradient can inform label growth about subtle differences in brightness that were lost during thresholding. The weight parameter in the convex combination allows the user to balance shape with brightness information.

After the first watershed segmentation, we are left with an over-segmentation. The core insight into how a human expert segments is now modeled by the *constriction factor* c: Each label remembers its radius at the thickest spot, i.e. the maximum of the distance transform within that label. At the interface between two neighboring labels, we scan for the maximum of the distance transform along the interface. This yields the local thickness of both labels at their border. Now, if this thickness value is close to the bigger of the two involved maxima, then the labels don't constrict much at their interface, and we assume that they actually belong to the same biological cell. Hence we merge them. Exactly how much of a constriction should warrant keeping both labels? This is another parameter to be specified. In order to keep the parameter scale-free, we model it as the ratio

$$c = \frac{\text{distance maximum at border}}{\text{distance maximum within both labels}}$$

From our experience, keeping labels separate for $c < 0.75$ seems to work best.

Even after merging labels, there might be structures present that are too small in volume to reflect biological cells. An effective way to merge those tiny labels with their bigger neighbors is to discard them and use the result as the seed for another iteration of watershed (using the same height map as before). This grows the remaining labels, filling the holes just created. The result is a labeled image, with different labels for connected components of the background, labels for individual cells wherever their shape allows for separation, and with a single label per cluster of cells where they are packed tightly.

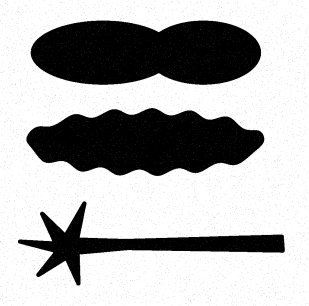
4. Runtime & Complexity

The building blocks of our method are thresholding, connected component search, distance transform, local maxima search, gradient magnitude, and watershed. All of these components have implementations with linear complexity $\mathcal{O}(n)$ where n is the number of voxels in the data set [4,6,9].

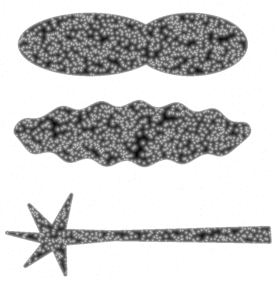
We have implemented our algorithm in a Jupyter notebook, employing numpy, Skimage and numba for efficient computation. With this approach, running the method on a 256^3 image takes < 5 minutes on an old laptop (Intel Core i3-3110M with 8GB RAM), including user input.

5. Validation

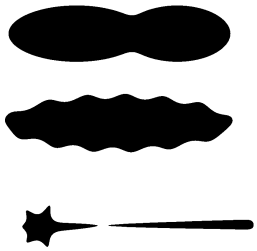
To assess the quality of the resulting segmentation, we apply it to a volumetric X-ray micro-tomography scan of a poplar leaf (see Figure 1). This image was downsized to get an isotropic voxel edge length of $0.325\mu m$, and then



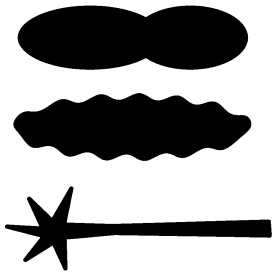
(a) Example of a thresholded source image. Note that there is salt-and-pepper noise present. Best viewed magnified.



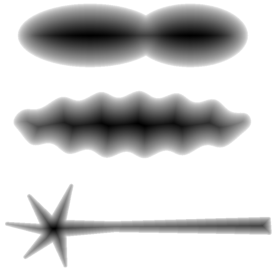
(b) The distance transform of Fig. 2a heavily distorted by the noise.



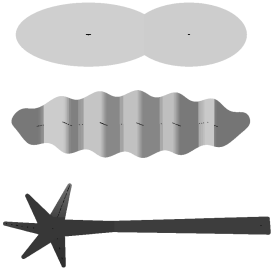
(c) Blurring the source image and thresholding removes the noise, but also details from the image. Note how the bottom shape becomes disconnected.



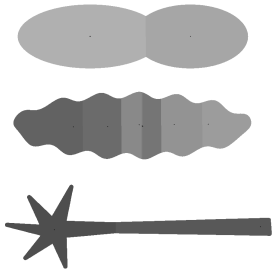
(d) Removing tiny connected components from both background and foreground in Fig. 2a cleans noise without losing image details.



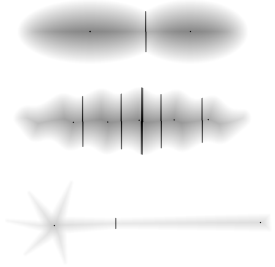
(e) The resulting distance transform of Fig. 2d



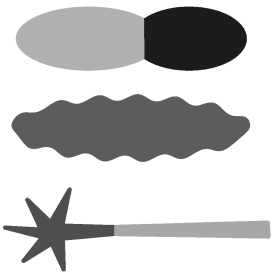
(f) Local maxima of the distance transform are marked as black dots. They are the used as seeds for watershed, yielding this oversegmentation.



(g) Blurring the distance transform prior to local maxima detection merges neighboring peaks, reducing oversegmentation and computation effort in subsequent steps.



(h) We investigate the diameter of Fig. 2g's region borders (marked as black lines): In the top shape, the distance maximum on the border between the two regions is far smaller than the maxima in either cell. This characterizes a noticeable constriction in the shape.



(i) The result after merging neighboring labels with only slight constrictions at their border.

Figure 2. 2d illustration of steps involved in the algorithm

cropped to 256^3 voxels. Ground truths of all cells present in three paradermal and four transversal 2d slices were hand labeled, with the exception of the within-vein cells which are too densely packed and were assigned a single label.

To compare the human-generated ground truth with the automated segmentation, we use the metric

$$\text{error} = 1 - F_1 = 1 - \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

suggested in [1] as error measure, as well as the information-based measures described in [11]: The variation of information (*voi*) of a segmentation with respect to the ground truth can be understood as a measure of oversegmentation, whereas voi of the ground truth with respect to another segmentation quantifies under-segmentation. We refer to these measures as *splits* and *merges*, respectively. From the results listed in Table 1, we recognize that both

over- and under-segmentation occurs, but mostly in moderate less-than-one-bit amount. Error values are high in the palisade (slice 5) due to low recall, even though the segmentation looks promising upon visual inspection. Correct segmentation of the water vein from its neighboring cells proves difficult for our shape-based approach, as can be observed on slices 6 and 7.

6. Conclusion

We have demonstrated a linear-time method for unsupervised seeded watershed segmentation of images in arbitrary dimension. Human interaction is required only to select a few parameters. The seeds for seeded watershed segmentation are auto-generated.

The novelty of our method compared to established watershed segmentation methods [7, 10, 15] lies in the following aspects:

1. A careful morphological pre-processing regime to compensate for the distance transform's sensitivity to noise.
2. Operating on level sets of the distance transform instead of explicitly constructing a shape skeleton [5, 12, 13].
3. The height map employed during watershed uses information from both image gradient and the shape of the foreground-background surface.
4. Over-segmentation resulting from watershed is corrected using a scale-independent, isotropic shape criterion that models human expert behavior.

7. Future Work

We intend to explore further shape-based instance segmentation methods and aim to improve the quality of our result. The local variation of the constriction factor near label borders may offer another suitable criterion for merging labels and correcting for watershed over-segmentation.

8. Acknowledgments

We acknowledge the Paul Scherrer Institut, Villigen, Switzerland for provision of beamtime at the TOMCAT beamline of the Swiss Light Source. The computational results have been partially achieved using the Vienna Scientific Cluster (VSC). This work was supported by the Vienna Science and Technology Fund (WWTF) project LS19-013 and by the Austrian Science Fund (FWF) projects M2245 and P30275.

References

[1] Ignacio Arganda-Carreras, Srinivas C. Turaga, Daniel R. Berger, Dan Cireşan, Alessandro Giusti, Luca M. Gambardella, Jürgen Schmidhuber, Dmitry Laptev, Sarvesh Dwivedi, Joachim M. Buhmann, Ting Liu, Mojtaba Seyedhosseini, Tolga Tasdizen, Lee Kamentsky, Radim Burget, Vaclav Uher, Xiao Tan, Changming Sun, Tuan D. Pham, Erhan Bas, Mustafa G. Uzunbas, Albert Cardona, Johannes Schindelin, and H. Sebastian Seung. Crowdsourcing the creation of image segmentation algorithms for connectomics. *Frontiers in Neuroanatomy*, 9, 2015.

[2] Dominique Attali, Jean-Daniel Boissonnat, and Herbert Edelsbrunner. Stability and computation of medial axes: a state-of-the-art report. *Mathematical Foundations of Scientific Visualization, Computer Graphics, and Massive Data Exploration*, 01 2009.

[3] Serge Beucher and Christian Lantuéjoul. Use of watersheds in contour detection. volume 132, 01 1979.

[4] Pedro Felzenszwalb and Daniel Huttenlocher. Distance transforms of sampled functions. *Theory of Computing*, 8, 08 2004.

[5] Mathieu Gaillard, Chenyong Miao, James Schnable, and Bedrich Benes. *Sorghum Segmentation by Skeleton Extraction*, pages 296–311. 01 2020.

[6] Lifeng He, Xiwei Ren, Qihang Gao, Xiao Zhao, Bin Yao, and Yuyan Chao. The connected-component labeling problem: A review of state-of-the-art algorithms. *Pattern Recognition*, 70:25–43, 2017.

[7] Wala'a Jasim and Rana Mohammed. A survey on segmentation techniques for image processing. *Iraqi Journal for Electrical and Electronic Engineering*, 17:73–93, 12 2021.

[8] Wenbo Jiang, Lehui Wu, Shihui Liu, and Min Liu. Cnn-based two-stage cell segmentation improves plant cell tracking. *Pattern Recognition Letters*, 128:311–317, 2019.

[9] Anton Kornilov and Ilia Safonov. An overview of watershed algorithm implementations in open source libraries. *Journal of Imaging*, 4:123, 10 2018.

[10] Aurélie Leborgne, Julien Mille, and Laure Tougne. Extracting noise-resistant skeleton on digital shapes for graph matching. pages 293–302, 12 2014.

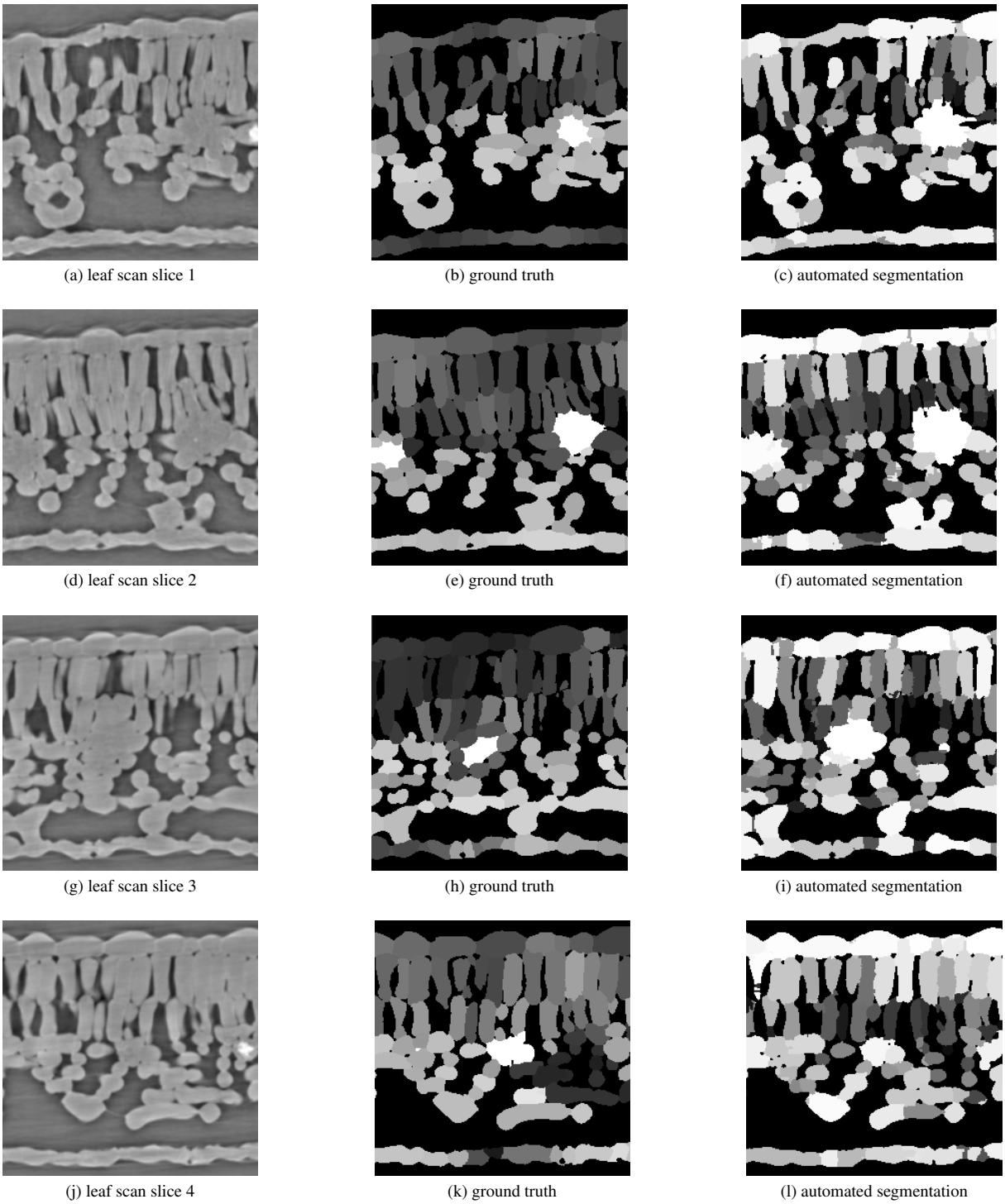
[11] Marina Meilă. Comparing clusterings—an information based distance. *Journal of Multivariate Analysis*, 98(5):873–895, 2007.

[12] Xiaopeng Sun, J. Pan, and Xiaopeng Wei. 3d mesh skeleton extraction using prominent segmentation. *Comput. Sci. Inf. Syst.*, 7:63–74, 02 2010.

[13] Seung tak Noh, Kenichi Takahashi, Masahiko Adachi, and Takeo Igarashi. Skelseg: Segmentation and rigging of raw-scanned 3d volume with user-specified skeleton. In *Graphics Interface*, 2019.

[14] Adrian Wolny, Lorenzo Cerrone, Athul Vijayan, Rachele Tofanelli, Amaya Vilches-Barro, Marion Louveaux, Christian Wenzl, Soeren Strauss, David Wilson-Sánchez, Rena Lymbouridou, Susanne Steigleder, Constantin Pape, Alberto Bailoni, Salva Duran-Nebreda, George Bassel, Jan Lohmann, Miltos Tsiantis, Fred Hamprecht, Kay Schneitz, and Anna Kreshuk. Accurate and versatile 3d segmentation of plant tissues at cellular resolution. *eLife*, 9, 07 2020.

[15] Nida M. Zaitoun and Musbah J. Aqel. Survey on image segmentation techniques. *Procedia Computer Science*, 65:797–806, 2015. International Conference on Communications, management, and Information technology (ICCMIT'2015).



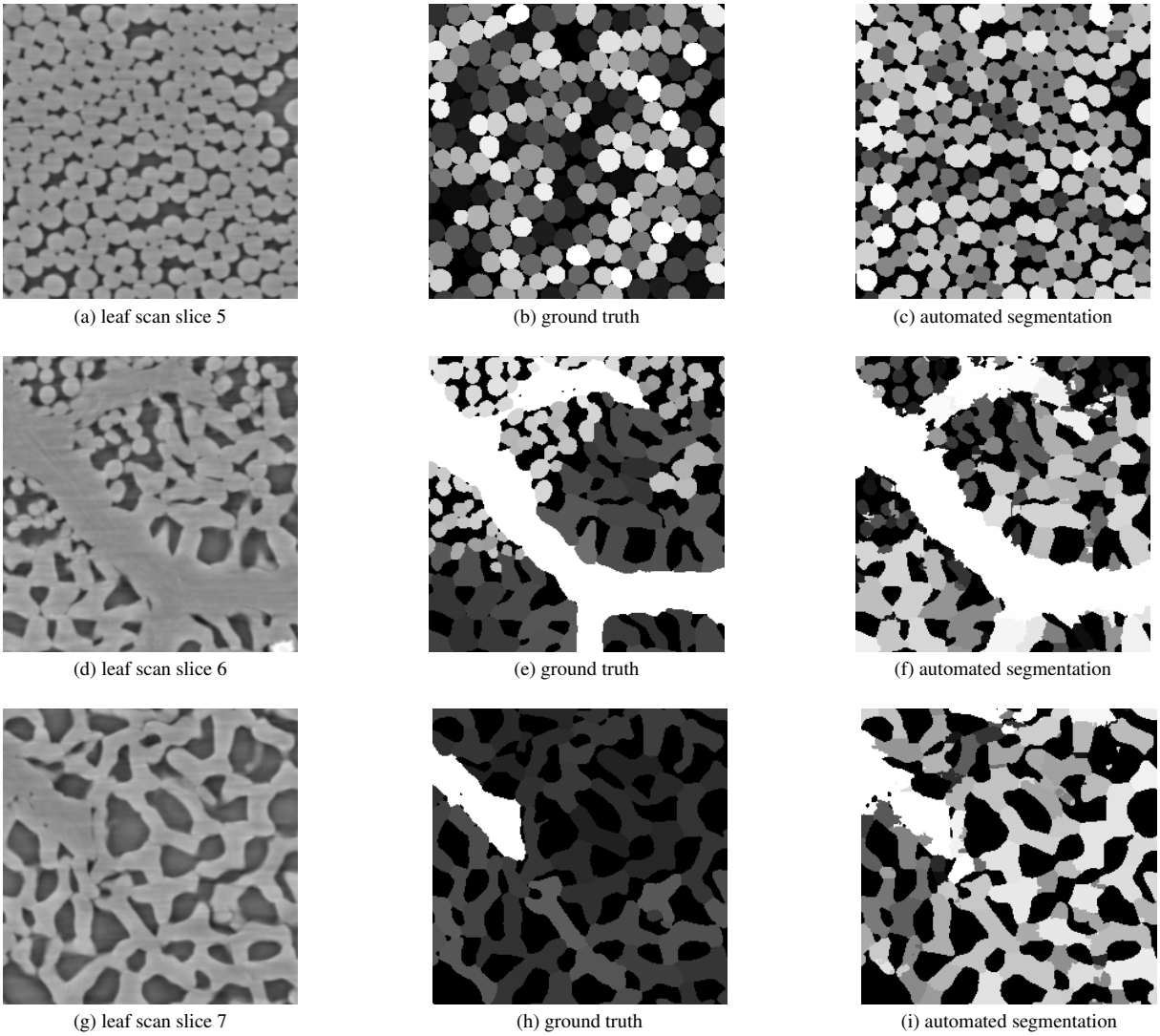
(a) leaf scan slice 1 (b) ground truth (c) automated segmentation

(d) leaf scan slice 2 (e) ground truth (f) automated segmentation

(g) leaf scan slice 3 (h) ground truth (i) automated segmentation

(j) leaf scan slice 4 (k) ground truth (l) automated segmentation

Figure 3. Transversal slices (perpendicular to the surface and to the leaf midrib)



(a) leaf scan slice 5 (b) ground truth (c) automated segmentation

(d) leaf scan slice 6 (e) ground truth (f) automated segmentation

(g) leaf scan slice 7 (h) ground truth (i) automated segmentation

Figure 4. Paradermal slices (parallel to the leaf surface)

slice no.	error [%]	precision [%]	recall [%]	splits [bit]	merges [bit]
1	6.3	98.8	89.0	0.6	0.7
2	4.5	98.0	93.1	0.7	0.5
3	12.2	88.6	87.0	0.8	0.8
4	14.3	91.8	80.4	0.9	0.8
5	40.1	93.0	44.1	0.6	1.2
6	37.6	52.9	76.0	1.5	0.9
7	15.9	88.2	80.4	1.2	0.7

Table 1. Segmentation error metrics