Efficient Semiautomatic Segmentation of Plant Biological Objects Based on Live-wire Methods

Wolfram Schoor
Fraunhofer Institute for Factory Operation and Automation (IFF)
Sandtorstrasse 22,
39106 Magdeburg, Germany
wolfram.schoor@iff.fraunhofer.de

Thomas Seidl
Fraunhofer Institute for Factory Operation and Automation (IFF)
Sandtorstrasse 22,
39106 Magdeburg, Germany
thomas.seidl@iff.fraunhofer.de

Felix Bollenbeck
Fraunhofer Institute for Factory Operation and Automation (IFF)
Sandtorstrasse 22,
39106 Magdeburg, Germany
felix.bollenbeck@iff.fraunhofer.de

Udo Seiffert
Fraunhofer Institute for Factory Operation and Automation (IFF)
Sandtorstrasse 22,
39106 Magdeburg, Germany
udo.seiffert@iff.fraunhofer.de

Bernhard Preim
Otto von Guericke University,
Universitätsplatz 2,
39106 Magdeburg, Germany
preim@isg.cs.uni-magdeburg.de

Rüdiger Mecke
Fraunhofer Institute for Factory Operation and Automation (IFF)
Sandtorstrasse 22,
39106 Magdeburg, Germany
ruediger.mecke@iff.fraunhofer.de

ABSTRACT

This paper presents a novel method for efficient semiautomatic multi-label segmentation of plant biological image data. The approach extends live-wire methods in order to facilitate exact user-steered segmentations for atlas generation. By integrating a segmentation-specific user interaction model into the live-wire formulation i) more exact segmentations, ii) increased computational efficiency, iii) without loss of generality are achieved. The concept of mutual influence of image feature based path costs and user input uncertainty are consistently combined. By incorporating user behavior into cost based delineation a more intuitive user interface is obtained also yielding in a more accurate segmentation. We introduce path-based methodologies, specific user interaction models and propose the combination of both of them. The purposefulness of the method is shown in an application comprising segmentation of histological section data supporting the generation of 3-D atlases.

Keywords: computer vision, user interaction, live-wire.

1 INTRODUCTION

Image segmentation is the basic principle behind various image processing tasks. The main goal is to obtain a higher level of information than a purely numerical representation of an image. The pixels are therefore combined into semantically equivalent groups [Tön05]. Segmentation consists of two steps, recognizing relevant objects in an image and delineating exact spatial dimensions and their features [FUM00]. Different algorithms that can automatically segment plant biological image data have been introduced for this non-trivial task [BS08] or [DBS+08]. These algorithms are generally based on existing reference data. Such model data is not available to generate atlases of unexplored plant biological histological images and must therefore either be segmented manually or semiautomatically. A set of semiautomatically segmented image slices (approximately 5% of a serial section image stack) can replace the purely manual segmentation of plant biological image data as presented in [GDB+07]. Image data consists of previously sectioned plant biological objects, i.e. caryopses of the Hordeum Vulgare, digitized with a color CCD camera with a resolution of 1600 x 1200 pixels under a light microscope. To facilitate their processing, the digitized images were converted to an 8 bit gray scale representation without any significant loss of relevant information (see [SBH+08]). Figure 1 is an image for such a segmentation task. The detail of the segmented region bounded in blue reveals obvious differences in the contour's characteristics (red and green box).

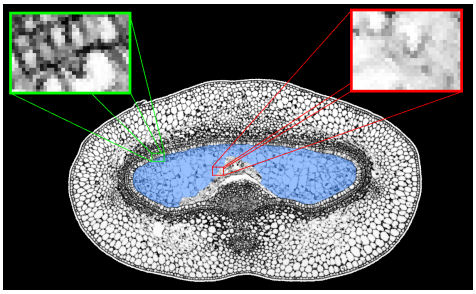
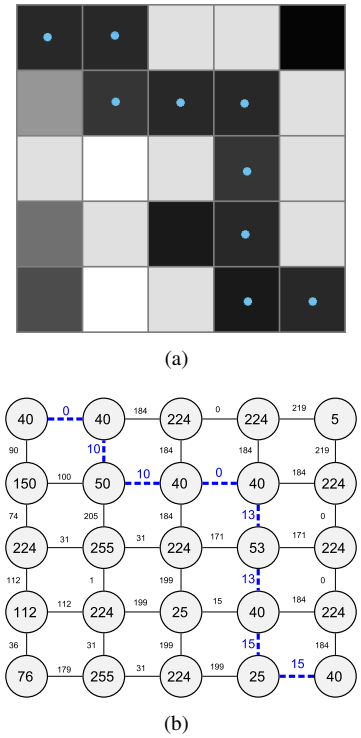


Figure 1: Slice of *Hordeum Vulgare* with different tissue characteristics (red and green box).

2 RELATED WORK

A graph-based semiautomatic interactive segmentation method, namely *live-wire*, was introduced by MORTENSEN et al. [MMBU92], [MB95], [BM97] and is related to *geometric active contours* presented by KASS et al. [KWT87]. The key feature of live-wire methods is their exact extraction of a boundary segment between a start point P_s and an end point P_e indicated by the position of the input device. The task of segmenting an entire region is broken down into optimal delineation of several parts comprising the entire contour. To do so, a cost map is calculated based on each pixel edge's properties and interpretation. A graph is introduced reflecting the cost map, which allows a path search for optimal path computation by the minimal accumulated path costs (cf. [HNR68] A*-algorithm). A set of weighted parameters derived from the pixel's neighborhood is employed to compute the path between the start and the end point. In an ideal case, a rough sketch of the contour suffices to extract the segment boundaries. Figure 2 illustrates the principle of the live-wire method. For clarity's sake, only a 4-cell neighborhood was selected for the visualization.



(a) (b)

Figure 2: (a) portion of a gray scale image, (b) shows respectively the graph, resulting from intensity differences. The intensity values appear in the nodes, related cost components between the nodes. Blue points indicate the optimal connection along the lowest total cost components (dashed line). The path follows the dark pixels from the upper left to the lower right corner.

3 METHODS

A core element of the live-wire method is the calculation of the aforementioned costs related to the edges in the graph. Depending on the image data, different cost factors can be applied for good segmentation results. The following listed cost factors constitute a combination of different cost factors, which are suitable for the used image data. The cost factors are derived from [MMBU92], [MB95] and are also based on own formulations.

The impact of different cost factors is weighted as an overall cost function. Let $\boldsymbol{p}$ and $\boldsymbol{q}$ be two adjacent pixels in an image slice S. The costs for a connection from $\boldsymbol{p}$ to $\boldsymbol{q}$ can be determined by a cost function $C(\boldsymbol{p},\boldsymbol{q})$, rendered thusly:

$$C(\boldsymbol{p},\boldsymbol{q}) = \sum_{n=0}^{N} \omega_n C_n(\boldsymbol{p},\boldsymbol{q}), \tag{1}$$

where N denotes the count of the different cost factors, ω_n the respective weighting factors and $C_n(\boldsymbol{p},\boldsymbol{q})$ the value of the n-th cost component between $\boldsymbol{p}$ and $\boldsymbol{q}$. The cost factors C_n can be differentiated as *static* and *dynamic* factors:

Static factors are calculated before segmentation and are independent of user interaction. Static cost factors indicate the cost to directly connect one pixel to its m neighbors in the graph-based path search. Filter operations are applied to determine the static cost factors (see Table 1):

Cost Factor	Cost Identifier (C_n)
Laplacian	C_L
Laplacian of Gaussian	C_{LoG}
Gradient magnitude	C_{GM}
Roberts Cross	C_{RX}
Difference of Intensity Values	$C_{\Delta I}$
Brightness Feature	C_{BF}

Table 1: Components of local cost functions between adjacent pixels.

An individual local cost factor is calculated following Equations 2-10:

$$C_L(p) = 1 - (L_{kernel}(x,y) \cdot S) \tag{2}$$

$$L_{kernel} = \frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} \tag{3}$$

were each summand in L_{kernel} stands for the second partial derivation in x respectively in y direction.

$$C_{LoG}(p) = 1 - (LoG_{kernel}(x,y) \cdot S) \tag{4}$$

$$LoG_{kernel} = -\frac{1}{\pi\sigma^4}\left(1 - \frac{x^2+y^2}{2\sigma^2}\right) e^{-((x^2+y^2)/2\sigma^2)} \tag{5}$$

 ISBN 978-80-86943-94-7

$$C_{RX}(p) = 1 - (RX_{kernel}(x,y) \cdot S) \tag{6}$$

$$RX_{kernel} = \frac{\partial p}{\partial x} + \frac{\partial p}{\partial y} \tag{7}$$

$$C_{GM}(q) = 1 - \frac{1}{max(G)} \sqrt{\left(\frac{dS(x,y)}{dx}\right)^2 + \left(\frac{dS(x,y)}{dy}\right)^2}, \tag{8}$$

$max(G)$ represents the largest gradient magnitude in the image. The $C_{\Delta I}(p)$ cost can be calculated as follows:

$$C_{\Delta I}(p) = \frac{|I(p) - I(q)|}{2^B - 1}, \tag{9}$$

where $I(p)$ and $I(q)$ denote the intensity value of their respective pixel positions and B the color depth of the current image. The brightness cost function C_{BF} can be expressed as:

$$C_{BF}(p) = I/max(I), \tag{10}$$

where low intensity values I correspond to low costs C_{BF}.
The *dynamic costs* cannot be computed in advance of a path search. They are computed during segmentation and reliant on user input. Table 2 lists the dynamic costs.

Cost Component	Cost Identifier (C_n)
L_0 distance to direct path	C_{L0}
L_1 distance to end point	$C_{\Delta P}$

Table 2: Dynamic components of the cost function.

Equation 11 calculates the orthonormal vector $\vec{v}$ to the straight line between P_s and P_e.

$$\vec{v} = \frac{\begin{pmatrix} P_{e_y} - P_{s_y} \\ -(P_{e_x} - P_{s_x}) \end{pmatrix}}{\sqrt{(P_{e_x} - P_{s_x})^2 + (P_{e_y} - P_{s_y})^2}} \tag{11}$$

Equation 12 determines the vector $\vec{r}$ from the start point P_s to the present location q.

$$\vec{r} = \begin{pmatrix} P_{q_x - s_x} \\ P_{q_y - s_y} \end{pmatrix} \tag{12}$$

Projecting $\vec{r}$ onto the normalized vector $\vec{v}$ yields the L_0 distance from point q to the straight line spanned by P_s and P_e and called $d_{L_0}(\boldsymbol{q})$ (see Equation 13).

$$d_{L_0}(\boldsymbol{q}) = \vec{v} \cdot \vec{r} = v_x\, r_x + v_y\, r_y \tag{13}$$

The costs increase as the pixel's distance to the ideal connection path increases (see Equation 14).

$$C_{L0}(P_s, P_e, q) = g(d_{L_0}(\boldsymbol{q})) \tag{14}$$

The function $g(x)$ may express this influence in a variety of ways, e.g. as a linear dependency or square root. Equation 15 provides an appropriate method:

$$g(x) = a\, x^2 \tag{15}$$

This quadratic function penalizes small deviations from the ideal line with small cost values, whereas larger distances become comparatively expensive. The increasing coefficient a of 0.25 damps too high cost values for distant input device positions.

The cost function $C_{\Delta P}$ represents the costs of a pixel q emerging from the distance to the end point P_e in relation to the neighboring pixel's p distance to P_e (see Equation 17). The term $d_{\Delta P}(i)$ with $i \in \{p,\ q\}$ denotes the L_1 distance (Manhattan distance) to P_e and is calculated with Equation 16.

$$d_{\Delta P}(i) = (P_{e_x} - i_x) + (P_{e_y} - i_y) \tag{16}$$

$$C_{\Delta P} = \begin{cases} 0.0, & \text{if } d_{\Delta P}(p) - d_{\Delta P}(q) \geq 1 \\ 0.5, & \text{if } d_{\Delta P}(p) - d_{\Delta P}(q) = 0 \\ 1.0, & \text{if } d_{\Delta P}(p) - d_{\Delta P}(q) \leq -1 \end{cases} \tag{17}$$

The basic live-wire methods may be extended in a variety of ways, e.g. by extending the search space to the third dimension [SPP00], [SPP01], [SUG+06] or by extracting the segmentation parameters for segments or segment parts [FUS+98] or [EKS92].
A detailed discussion of these extension is out of the scope of this paper. For a detailed analysis of live-wire's implementation and interaction methods for image segmentation, the authors refer to [BM97] or [OS01].

4 EXTENDING THE LIVE-WIRE METHOD

In [SBH+08], SCHOOR et al. introduced interaction-based support of segmentation, i.e. *speed-dependent automatic zooming* (SDAZ) based on IGARASHI [IH00] and *pseudohaptic feedback* (see LÉCUYER [LBE04]), to improve the segmentation process in terms of segmentation speed and accuracy of results. A user study demonstrated that this also works in practical use [SBH+08].
Different application-dependent algorithms and heuristics accelerate the run- and reaction-time of the application presented here.

4.1 Restricting the Search-Space

Apart from the large quantity of image data, the main disadvantage is that only a fraction of the data is relevant to the determination of the correct path. A

 ISBN 978-80-86943-94-7

combination of new and established strategies computes paths faster and supports real-time interaction.

Decreasing the Global Search-Space:
In this very important step, the information space must be narrowed before calculating the path, even before generating the cost map. This can be done by simply thresholding the image and ignoring undesired intensity values in image parts or pixels.
Due to the nature of the image data approximately 50% of the pixel amount can so be masked in advance of all further computational steps. Unimportant areas (black) can be masked as in Figure 3.

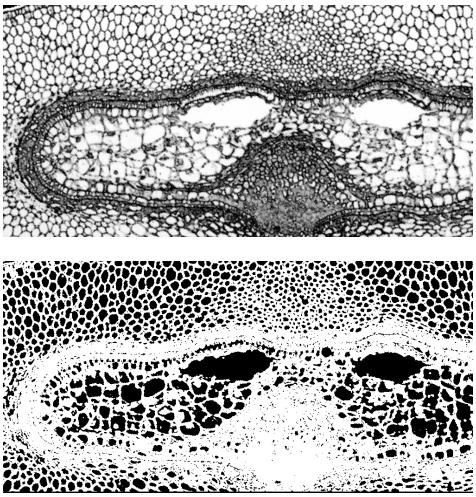


Figure 3: Original (top) and masked (bottom) image slice.

On the Fly Path Restriction:
The segmentation of object boundaries can be difficult, especially if the boundaries are discontinuous and very close to each other (see Figure 1). Manually or automatically setting a user-defined control point in the system restricts the search space to a rectangular path search between the two consecutive points and a predefined pixel distance to the direct path connection. Unlike [SPP00] where the target search space can broadly be masked before segmentation starts, this method restricts the path on the fly between the last segment and the new current point. Figure 4 presents the principle. The yellow border encloses the search space with the present distance parameter.
This restriction is useful because a pixel's corresponding node is not expanded in the graph if its distance to the ideal line exceeds the defined value. This limits the depth of a path search and allows it to be terminated faster. The expected speed increase will be small in regions with strong straight edges but will prevent the bleeding of the contour in homogeneous regions with weak edges. Selecting a small distance value accelerates a path search (very small information search space) but attracts the contour line to the ideal line between the two considered points and can therefore differ from the required values.

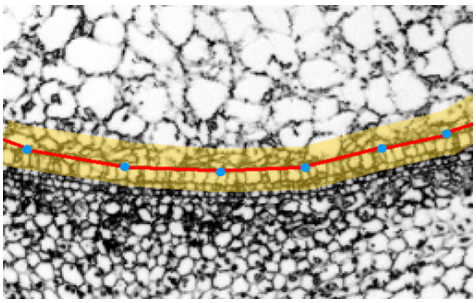


Figure 4: A border (with predefined pixel distance to the ideal line between two consecutive points) is used to restrict the information search space for the path search.

Speed Dependent Path Restriction:
The speed of user interaction during segmentation is a characteristic feature of segmentation (see [SBH+08]) that can be used to determine whether homogeneous regions or curvy contours must be segmented. This allows determining the distance value automatically. If user interaction during segmentation is slow, the distance value should be decreased.

Masking of Regions
Recalling LUCCHESE [LM01], that the segmentation is a subdivision of the image into n disjunct sections, a sequentially performed segmentation of two adjacent regions can cause the formation of "isles", due to filter operations, user inaccuracies or other reasons. To avoid this and speed up the segmentation process a masking of already segmented regions ensures an ideal segmentation of the image (one pixel only belongs to one region). If a user navigates into one masked region, the pixel adjacent to that regions's boundary is selected as the current point for path extraction. Figure 5 presents an example of a new path segment (blue) adapting to the region masked in red.

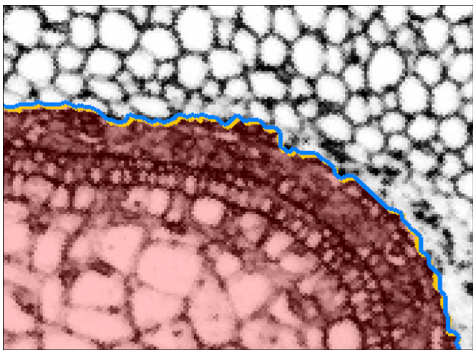


Figure 5: The masked region (transparent red) cannot be segmented again.

Tiling of the Image:
The mass of histological image data make it impossible

to guarantee that the different subsequent image based filter operations are calculated efficiently, especially when other time-critical calculations must be solved synchronously. One solution is to use graphics hardware as numerical processor as outlined in [RVC07]. The drawback of this solution is the need for appropriate graphics hardware in the meaning of limited texture size. Another solution is to break down an image segmentation problem into smaller elements, e.g. by using overlapping tiles as demonstrated in CRISP et al. [CPR03]. The overlap of the tiles depends on the maximum filter size used. An example is presented in Figure 6.
The graph segment corresponding to a tile is generated when:

1. The input device enters a new tile
2. The path search arrives a new tile
3. Enough time for further tile computation is left (concerning the real time interaction criterion)

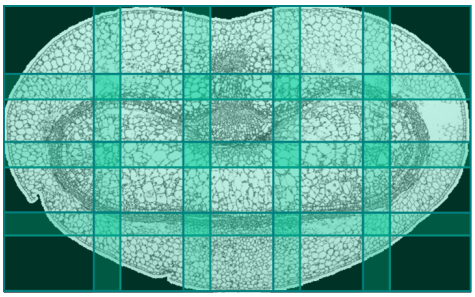


Figure 6: Tiling a large image slice with overlap.

4.2 Improving User Interaction

The system's real time response speed during segmentation can be assumed. More important is a simple interaction mechanism that supports users during the time consuming task of segmentation.

Automatic End Point Shifting:
The contour of the region between the last point and the input device's current position is calculated during semiautomatic segmentation. Control points and input device positions are entirely bound to user interaction. The connection between these points can be calculated very efficiently. Control point position and end point position must therefore be placed exactly. Manually setting control points can be time consuming and entails additional user effort. A possible end point is set orthogonally to the line between the last control position and the input device's current position (see Figure 7).

Automatic and Manual Control Points:
The complexity of path calculation depends directly

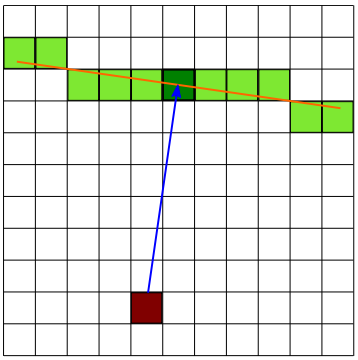


Figure 7: Scheme to calculate potential end points.

on the path length. Are start and end points for the path calculation further away from each other, more pixel have to be taken into account for the correct path determination. A divide and conquer strategy is employed to accelerate path determination. The user roughly sketches the contour to segment. If the input device position exceeds a predefined distance Δ_d to the start position or a predefined time interval Δ_t, a control point is automatically added to the position. The path determined becomes fixed.
The path determination for the whole path will be calculated in the following only between the last control point and the end point respectively the next automatically or manually inserted control point.
Figure 8 presents a segment part (red dashed line) following a segment boundary. Green points are control points manually inserted in difficult positions in the image (e.g. intersecting lines, line breaks, sharp changes of direction, etc.). Purple points are cooled control points that have remained unchanged for a long time (see [BM96] for cooled control points) or have a distinct number of following control points and therefore, cannot be deleted by incidence (for example by doing a segmentation loop). Blue points are recently inserted control points. These points can be deleted by entering a point surrounding region with a predefined radius. Purple points can then warm up again, i.e. turn blue again.

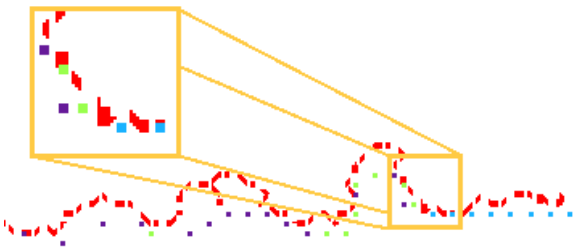


Figure 8: Temporary segment part with manually inserted control points (green), automatically inserted break points (blue) and locked break points (purple), also shown in closer detail.

Disabling the Path Search:
Simply deactivating a path search may be advantageous in some cases, e.g. if the contour line changes radically over time. The parameters initially used to extract the contour may result in erroneous segmentation results. Thus, the segmentation can be partially done manually by contemporaneously pressing a predefined key.

5 RESULTS

The algorithms presented were tested on histological plant biological data sets. The results from the segmentation of a representative number of image slices can be used for the method of automatic segmentation proposed by BOLLENBECK et al. [BS08].

5.1 Experience

In Figure 9 a complete segmentation of a histological slice is shown with respective tissues.

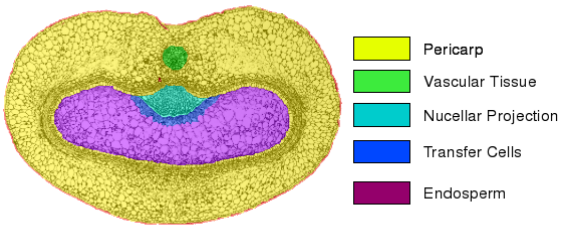


Figure 9: Labeled histologic image slice with legend of tissues.

Experimental studies evaluated the weights of the particular cost factors. The cost factors are mapped in the normalized interval $[0, 1]$. This enables a uniform weighting due to the weighting factors. Table 3 presents the different tissues of a grain caryopsis and their weighting factors. The weighting factors are bounded by the interval $[0, 100]$. These experimental evaluated values can be used for similar segmentation tasks of histological plant biological image data as initial values for the used cost factors. This can lead to a faster setting of the cost factors for well suited segmentation results.
Figure 10 presents a part of a reconstructed model and the respective sample segmentation slices.

5.2 User Tests

The developed live-wire method was used in practical tests for the segmentation of biological plant data. Users were asked to segment the *endosperm* of the barley grain, which is shown in Figure 9 (violet). This segment was chosen because its border is defined by different features, including strong clear edges but also

Tissue	C_L	C_{LoG}	C_{GM}	C_{BF}	C_{RX}	$C_{\Delta I}$	C_{L0}	$C_{\Delta P}$
Exterior	30	20	0	20	15	25	5	5
Vascular bundle	10	20	5	15	20	10	10	10
Pericarp	30	20	0	20	15	25	5	5
Transfer cells	20	15	10	20	15	20	5	5
Blowhole	25	25	5	10	10	20	5	5
Nucellar projection	30	25	0	20	10	10	5	5
Endosperm	30	25	0	20	15	10	10	5

Table 3: Weights of the different cost factors for each tissue.

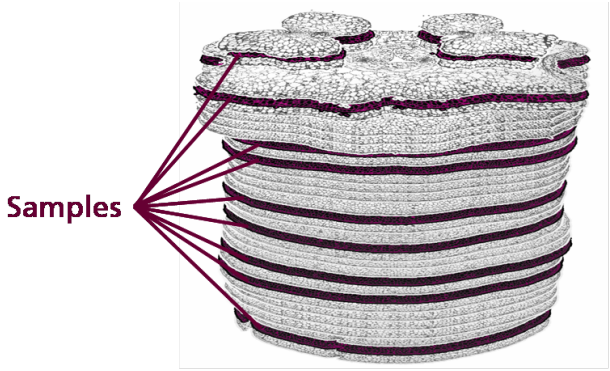


Figure 10: Reconstructed upper model part of a grain model using sample segmentations

weak unclear or even interrupted edges.
The users were also asked to segment the chosen part completely manually without system assistance. Eight out of ten users achieved faster results with the here proposed semiautomatic method. Partially the live-wire method was twice as fast as the manual segmentation. The two users, who needed more time declared that they did not use the option to switch to manually segmentation during the process.
This leads to time-consuming segmentation attempts by repeating wrong segmentations in some parts of the image with weak edges or large gaps between corresponding edges.

Accuracy:
According to MORTENSEN AND BARRETT [MB95] the accuracies of the users segmentations were compared with a segmentation, which was manually done by an expert (gold standard). The resulting accuracy graphs show a very high degree of similarity over all segmen-

tation tools which were tested. The commercial software solutions achieve a better correlation than the here provided software in the range of very small deviations to the gold standard (especially Photoshop). If the Hausdorff-distance is greater than 2 pixels the presented approach performs best.
With respect to physical diameters of a tissue boundary, which can be for example up to 4 pixels in size in the image data, Hausdorff-distances of 3 or 4 pixels are still within an acceptable accuracy range.

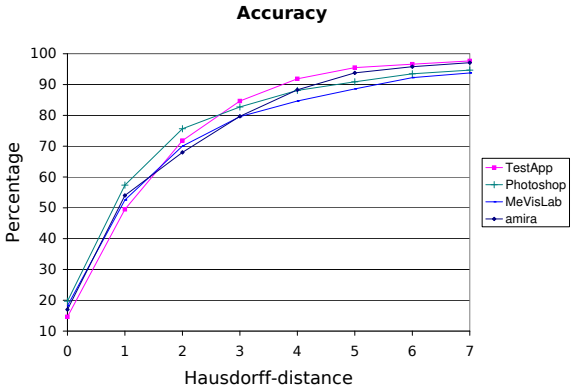


Figure 11: Accuracy results in comparison with other segmentation tools.

Reproducibility:
By comparing the results of different segmentations with each other the developed method had a high reproducibility in intra-user and inter-user comparison (see Figure 12 a) and Figure 12 b)). That means that different segmentations of one user were almost the same but also the segmentations of different user had a very high similarity.

Usability:
Asking the users for a subjective rating of the presented method, nine out of ten users confirmed a good usability of the presented method, which is easy and fast to learn. Especially the automatic placement and the different states of the control-points were appreciated by most users.

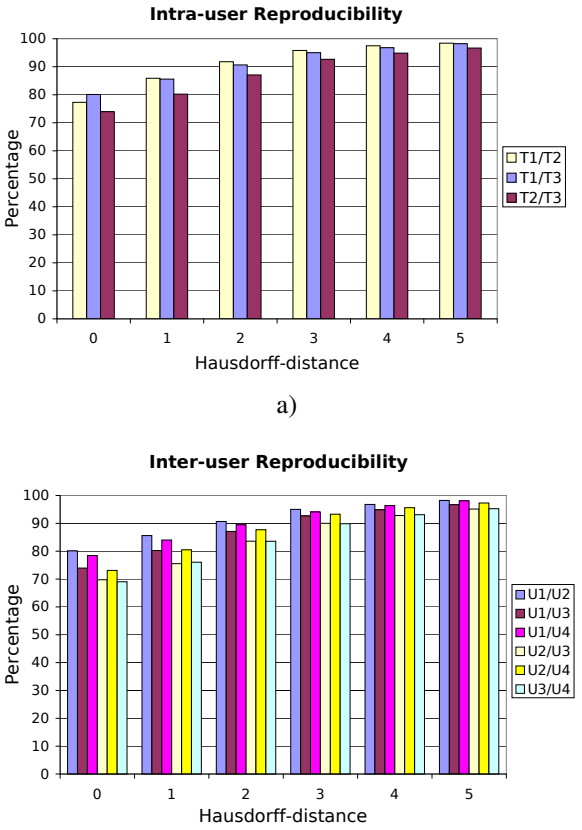


Figure 12: The reproducibility is shown on the basis of one histological image slice segmentation. The matching results of three segmentations $T1-T3$ of one user are shown in a) and the matching of segmentations of four different users $U1-U4$ in b). The X-Axis stands for the difference of the compared segmentation results in pixel (Hausdorff-distance). The ordinate indicates the amount of pixel which fulfill this criterion.

6 CONCLUSIONS

A variety of enhancements improve semiautomatic segmentation tasks on digitized plant biological serial sections. A user test demonstrated that the system is suitably usable. The objective improvements in comparison to commercial segmentation software must be further evaluated. The presented system has been proven to be as accurate as *amira* `www.amira.com`, *MeVisLab* `www.mevislab.de` and *Photoshop*'s `www.adobe.com` semiautomatic segmentation capabilities for selected image data.
The system was especially designed for large image slices. In contrast, standard medical applications (e.g. CT/NMRi) use typically less than a quarter of the here required image size for segmentation tasks.
The combination of mutual influence of image feature based path costs and user input uncertainty lead to faster and more accurate segmentations than manual segmentations. The incorporation of user behavior into a cost based delineation was perceived as intuitive by almost all users.
Furthermore, path-based methodologies and specific user interaction models were introduced and the usefulness of a combination of both of them were shown. The presented semiautomatic segmentation is capable of fast and accurate segmentations of plant biological sample data. Therefore, the results can be used as input for the automatic segmentation method proposed by BOLLENBECK et al. [BS08].
The presented paper constitutes a contribution to support the generation of plant biological 3-D atlases.

7 FUTURE DIRECTIONS

Further investigations will point toward the evaluation of the presented strategies' influence to the segmentation results. A user study must be done to validate the tendency of the performed user tests.
The results have to be further verified for different segmentation scenarios especially in the context of practical use.

8 ACKNOWLEDGEMENT

The Federal Ministry of Education and Research (BMBF) is supporting this research with the grants 0313821B and 0313821A. The authors would like to thank their colleagues from the IPK Gatersleben and their student assistants for their support and assistance.

9 REFERENCES

[BM96] W. A. Barrett and E. N. Mortensen. Fast, Accurate, and Reproducible Live-wire Boundary Extraction. In *VBC '96: Proceedings of the 4th International Conference on Visualization in Biomedical Computing*, pages 183–192, London, UK, 1996. Springer-Verlag.

[BM97] W. A. Barrett and E. N. Mortensen. Interactive Live-wire Boundary Extraction. *Medical Image Analysis*, 1:331–341, 1997.

[BS08] F. Bollenbeck and U. Seiffert. Fast Registration-based Automatic Segmentation of Serial Section Images for High-resolution 3-D Plant Seed Modeling. In *ISBI*, pages 352–355. IEEE, 2008.

[CPR03] D. J. Crisp, P. Perry, and N. J. Redding. Fast Segmentation of Large Images. In *ACSC '03: Proceedings of the 26th Australasian computer science conference*, pages 87–93, Darlinghurst, Australia, Australia, 2003. Australian Computer Society, Inc.

[DBS+08] V. J. Dercksen, C. Brüß, D. Stalling, S. Gubatz, U. Seiffert, and H.-C. Hege. Towards Automatic Generation of 3D Models of Biological Objects Based on Serial Sections. In L. Linsen, H. Hagen, and B. Hamann, editors, *Visualization in Medicine and Life Sciences*, pages 3–25. Springer-Verlag Berlin Heidelberg, 2008.

[EKS92] P. J. Elliott, J. M. Knapman, and W. Schlegel. Interactive Image Segmentation for Radiation Treatment Planning. *IBM Syst. J.*, 31(4):620–634, 1992.

[FUM00] A. X. Falcão, J. K. Udupa, and F. K. Miyazawa. An Ultra-fast User-steered Image Segmentation Paradigm: Live-wire-on-the-fly. *IEEE Trans. on Medical Imaging*, 19(1):55–62, January 2000.

[FUS+98] A. X. Falcão, J. K. Udupa, S. Samarasekera, S. Sharma, B. E. Hirsch, and R. A. Lotufo. User-steered Image Segmentation Paradigms: Live-wire and Live-lane. *Graphical Models and Image Processing*, 60(4):233–260, July 1998.

[GDB+07] S. Gubatz, V. J. Dercksen, C. Brüß, W. Weschke, and U. Wobus. Analysis of Barley (Hordeum Vulgare) Grain Development Using Three-dimensional Digital Models. *The Plant Journal*, 52(4):779–790, 2007.

[HNR68] P. E. Hart, N. J. Nilsson, and B. Raphael. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *Systems Science and Cybernetics, IEEE Transactions on*, 4(2):100–107, 1968.

[IH00] T. Igarashi and K. Hinckley. Speed-dependent Automatic Zooming for Browsing Large Documents. In *UIST '00: Proceedings of the 13th annual ACM symposium on User interface software and technology*, pages 139–148, New York, NY, USA, 2000. ACM Press.

[KWT87] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. In *Proceedings of the First International Conference on Computer Vision*, pages 259–268, June 1987.

[LBE04] A. Lécuyer, J.-M. Burkhardt, and L. Etienne. Feeling Bumps and Holes Without a Haptic Interface: The Perception of Pseudo-haptic Textures. In *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 239–246, New York, NY, USA, 2004. ACM Press.

[LM01] L. Lucchese and S. K. Mitra. Color Image Segmentation: A State-of-the-art Survey. In *Proceedings of the Indian National Science Academy*, pages 207–221, 2001.

[MB95] E. N. Mortensen and W. A. Barrett. Intelligent Scissors for Image Composition. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 191–198, New York, NY, USA, 1995. ACM.

[MMBU92] E. Mortensen, B. Morse, W. Barrett, and J. Udupa. Adaptive Boundary Detection Using "Live-wire" Two-Dimensional Dynamic Programming. *IEEE Proceedings Computers in Cardiology*, pages 635–638, October 1992.

[OS01] S. D. Olabarriaga and A. W. M. Smeulders. Interaction in the Segmentation of Medical Images: A Survey. *Medical Image Analysis*, 5(2):127 – 142, 2001.

[RVC07] Y. Roodt, W. Visser, and W. A. Clarke. Image Processing on the GPU: Implementing the Canny Edge Detection Algorithm. In *Proceedings of the Eighteenth International Symposium of the Pattern Recognition Association of South Africa (PRASA 2007)*, pages 45–50, 2007.

[SBH+08] W. Schoor, F. Bollenbeck, M. Hofmann, R. Mecke, U. Seiffert, and B. Preim. Automatic Zoom and Pseudo Haptics to Support Semiautomatic Segmentation Tasks. In V. Skala, editor, *16th WSCG 2008*, WSCG'2008 Full Papers Proceedings, pages 81–88. WSCG, University of West Bohemia, February 2008. Full Paper.

[SPP00] A. Schenk, G. P. M. Prause, and H.-O. Peitgen. Efficient Semiautomatic Segmentation of 3D Objects in Medical Images. In *MICCAI '00: Proceedings of the Third International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 186–195, London, UK, 2000. Springer-Verlag.

[SPP01] A. Schenk, G. Prause, and H.-O. Peitgen. Local Cost Computation for Efficient Segmentation of 3D Objects with Live Wire. In M. Sonka and K. M. Hanson, editors, *Medical Imaging 2001: Image Processing*, number 4322 in Proceedings of SPIE, pages 1357–1364, 2001.

[SUG+06] A. Souza, J. K. Udupa, G. Grevera, Y. Sun, D. Odhner, N. Suri, and M. D. Schnall. Iterative Live Wire and Live Snake: New User-steered 3D Image Segmentation Paradigms. In *SPIE Medical Imaging*, volume 6144, pages 1159–1165, 2006.

[Tön05] K. D. Tönnies. *Grundlagen der Bildverarbeitung*. Pearson Studium, München, May 2005. ISBN-10: 3827371554.