Joanna Sekulska-Nalewajko*, Jarosław Gocławski*

An Image Analysis Method for the Automatic Measurement of Selected Morphological Features of Wheat Shoots

1. Introduction

The aim of this work was to establish the method for evaluation of wheat shoot length and area on the basis of scanned images. The measurement of shoot size is a useful tool for estimation of severity of metal-induced stresses. The plant material considered in this study was cultivated in the Department of Plant Physiology and Biochemistry of University of Łódź and was the subject of experiments on toxic effects of heavy metals on young wheat organisms as well as beneficial effect of selenium. The plants were exposed to the chemical factors during first five weeks of cultivation. Building the algorithm for wheat shoot measurements has been a part of a series collaborative studies on cereal species. Previously the automated methods for root systems measurement [2] as well as the detection of root discolourations of different origins [3] have been prepared.

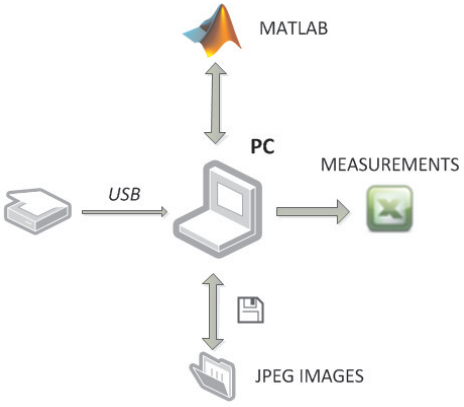


Fig. 1. The block diagram of wheat shoots image measurement system

* Computer Engineering Department, Technical University of Lodz, Poland

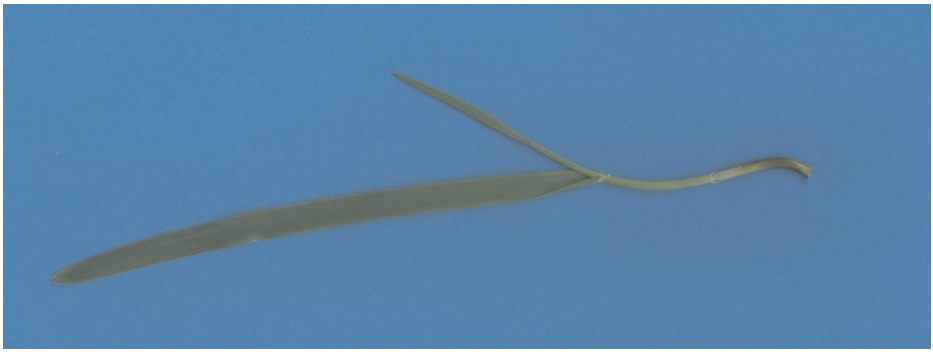
The measurement system acquisition is based on a flatbed scanner with *300 dpi* and *24 bit/pixel* resolutions, controlled by firmware software package. The scanner is connected via USB to a laptop equipped with dual core processor *Intel Core (TM) 2 Duo T5750 2 GHz*, *4 GB* RAM and operating system *Windows 7 32 bit* (Fig. 1).

The measurement algorithm has been developed in MATLAB 2010 environment also applying *Visual Studio Express 2008* C++ compiler to create selected, time-critical functions as MEX files, with the purpose to speed up their execution [14]. Obtained measurement results are sent to *Excel* application file to preserve them for further studies.

2. Wheat shoot objects segmentation

The examined wheat shoots are placed horizontally on a scanner glass so that the stem is always located on the right side of an acquired image (Fig. 2). The tested material comes from hydroponic cultures, so it has no soil pollutions. The shoots are always green or locally yellow-green. To facilitate image segmentation they are put on a uniformly coloured, highly saturated background with the colour different from any colour visible in plant tissue – typically blue as shown in Figure 2.

a)



b)

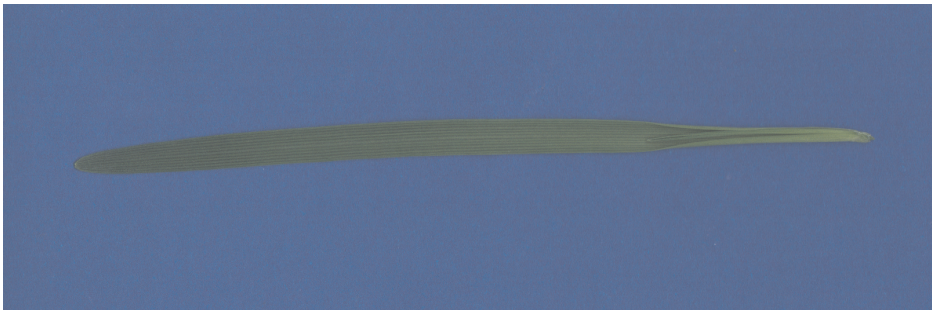


Fig. 2. The example images of wheat shoots underlying measurement on a blue background: a) typical shoot with two leaves; b) one-leaf shoot (the second leaf is small and hidden inside the main leaf blade)

Thanks to providing highly saturated background colours global thresholding of the hue image component I_H (Equ. (1)) has been selected for shoots segmentation [1]. The built-in library functions from *Image Processing Toolbox* have been used for this purpose [15].

$$\{I_H, I_S, I_V\} = rgb2hsv(I_{RGB}) \tag{1}$$

where:

I_{RGB} – an original colour leaf image,
I_H, I_S, I_V – hue, saturation an value images respectively.

The hue image thresholding is expressed by:

$$I_B = im2bw(I_H, graythresh(I_H)) \tag{2}$$

where *graythresh* computes the threshold value according to the Otsu method [12] and *im2bw* executes thresholding over I_H. The hue image component from definition is independent on illumination level, moreover even illumination is ensured at the scanning in the whole field of view, so the proposed global threshold is a reasonable solution to obtain the binary mask image I_B from a shoot image. An example segmentation result has been shown in Figure 3.

Fig. 3. The segmentation result example of shoot image from Figure 2a

The segmentation stage is finished with two "cleaning" procedures [13, 15]:

1) *imfill* (I_B, '*holes*') – to fill all potential holes that could appear inside of shoot object,
2) *bwareaopen* (I_B, A_{MIN}) – with the limited area A_{MIN} defining small junk objects to remove.

The segmentation divides the original image I_{RGB} in two parts: binary shoot object O_{B} marked as white and its black background.

3. Shoot object contouring

The shoot contour is used to find shoot object specific locations like the tip and the basal end, offshoot starting pixels and the tips of all leafs important for further image analysis. Optionally the contour can be smoothed before the skeletoning step of shoot object (Fig. 3) to avoid spurious skeleton branches, which don't represent leaf's medial axes and are induced by contour ripples. The contour is obtained by tracing 8-adjacent white edge pixels of the shoot object O_B, using right most looking rule [5]. The tracking function from MATLAB *Image Processing Toolbox* used for this purpose is shown in Equ. (3).

$$C = \left[(y_c, x_c)\right] = bwboundaries(I_B, N_b) \tag{3}$$

where:

C – the array of contour points (y_c, x_c),
$N_b = 8$ – means 8-connectivity of neighbour pixels.

The contour smoothing consists in the applying of *Elliptical Fourier Descriptors* (EFD) to limit contour frequencies [7]. The forward and inverse transform have been written as separate functions in MEX files complying with the Fourier series expansion rules:

$$\begin{aligned} x'_c &= a_0 + \sum_{n=1}^{N} a_n \cos\left(\frac{2\pi nt}{T}\right) + \sum_{n=1}^{N} b_n \sin\left(\frac{2\pi nt}{T}\right) \\ y'_c &= c_0 + \sum_{n=1}^{N} c_n \cos\left(\frac{2\pi nt}{T}\right) + \sum_{n=1}^{N} d_n \sin\left(\frac{2\pi nt}{T}\right) \end{aligned} \tag{4}$$

where:

T – the number of steps needed to traverse the entire contour,
t – the step required to traverse 1 pixel along the closed-contour C,
a_n, b_n, c_n, d_n – frequency coefficients truncated to the N of harmonics to create a smooth approximation of the boundary.

The rules for calculation of the coefficient values a_n, b_n, c_n, d_n are given e.g. in [11]. After EFD filtering the consecutive transformed pixels of the new contour $C(x_c', y_c')$ are connected with lines creating the rim of a new object which inside is then filled as a "hole"(*imfill*).

Unfortunately, especially immature leafs are very narrow at tips and include high contour frequencies, what suggests careful cutting of the harmonics and dense contour sampling to avoid post filtering deformations. With these assumptions EFD filtering is very time-consuming and therefore is regarded only as an option, which can be replaced in most cases by appropriate skeleton pruning. The illustration of zoomed shoot contours with and without EFD smoothing has been shown in Figure 4.

Fig. 4. The zoomed fragments of wheat shoot object contour after smoothing with EFD transform (first 64 harmonics). The contour is shown in white, the object in grey

4. The extraction and analysis of medial axes

The evaluation and tracing of medial axes plays a key role in the shoots measurement algorithm. The thinning procedure published in [8] has been selected, which is relatively robust to possible contour distortions. It is embedded in the following MATLAB function:

$$I_{SK} = bwmorph(I_B, 'thin', Inf) \tag{5}$$

where:

I_{SK} – binary image of shoot skeleton,

Inf – means stop thinning when no I_{SK} image changes occur between iterations.

The skeleton image I_{SK} obtained in Equ. (5) consists of white skeleton pixels and a black background. Before further analysis it should be changed to a graph form. This mapping includes the following three stages:

1) the identification of skeleton end nodes and internal nodes,
2) traversing white pixels between each end node and the other first-encountered node,
3) traversing white pixels from the internal nodes which 8-connected white neighbours are still unused.

p_4	p_3	p_2
p_5	p_0	p_1
p_6	p_7	p_8

Fig. 5. The 3×3 neighbourhood $N_b(p_0)$ of a white pixel p_0 in the skeleton image I_{SK}

Based on 3×3 pixel neighbourhood illustrated in Figure 5 end node has been defined as a white pixel p_0 satisfying the following conditions [6]:

$$\begin{aligned} &\sum_{i=1}^{8} I_{SK}(p_i) \le 2 \ \wedge \ \sum_{i=1}^{8} p_i^+ = 1 \quad \text{or} \\ &\sum_{i=1}^{8} I_{SK}(p_i) = 3 \ \wedge \ \sum_{i=2,4,\ldots}^{8} p_i^+ = 1 \end{aligned} \tag{6}$$

Additionally every internal node p_0 can be determined by:

$$\sum_{i=1}^{8} I_{SK}(p_i) > 2 \ \wedge \ \sum_{i=1}^{8} p_i^+ > 2 \tag{7}$$

In both cases $I_{SK}(p_i) \in \{0,1\}$, p_i^+ means positive slope ($I_{SK}(p_{i-1}) = 0 \wedge I_{SK}(p_i) = 1$) in a cyclical traversal through the neighbours of p_0 pixel.

The path $\mathbf{B}_i$ between two skeleton nodes is tracked in every neighbourhood $N_b(p_j)$ of currently labelled pixel p_j by selecting its successor p_{j+1}. This pixel is the nearest white neighbour $p_n = NN(p_j)$, for which $I_{SK}(p_n) = 1$ (firstly 4-connected neighbours). The path $\mathbf{B}_i$ represents a pixel array corresponding to the skeleton graph edge assigned to the initial tracking node. The tracking algorithm has been symbolically written in Figure 6.

$$\begin{aligned} &\textit{foreach} \quad i \in \{1, K\ , n_N\}, \\ &\quad p_1 = \mathbf{N}(i), \\ &\quad \mathbf{B}_i = [\,], \\ &\quad \textit{if} \quad I_{SK}(p_1) \neq 1, \\ &\qquad \textit{continue}, \\ &\quad \textit{foreach} \quad j \in \{1, 2, K\}, \\ &\qquad \mathbf{B}_i = \left[\mathbf{B}_i, p_j\right], \\ &\qquad \textit{if} \quad p_j \in \mathbf{N} \wedge p_j \neq p_1, \\ &\qquad\quad \textit{break}, \\ &\qquad I_{SK}(p_j) = 2, \\ &\qquad p_{j+1} = NN(p_j), \end{aligned}$$

Fig. 6. The pseudo-code snippet for the identification of skeleton branches; $\mathbf{B}_i$ – the array of i-th branch pixels, $\mathbf{N}$ – the array of skeleton nodes, n_N – the number of skeleton nodes, [ ] – empty array operator

The graph of shoot skeleton is finally stored as the array of branch starting nodes **N**, and the set of branch pixel arrays $\{\mathbf{B}_i: i = 1, ..., n_N\}$ associated with these nodes. Let **V** means the array of only the end nodes and $\{\mathbf{E}_i: i = 1, ..., n_V\}$ the set of respective end branches associated with them.

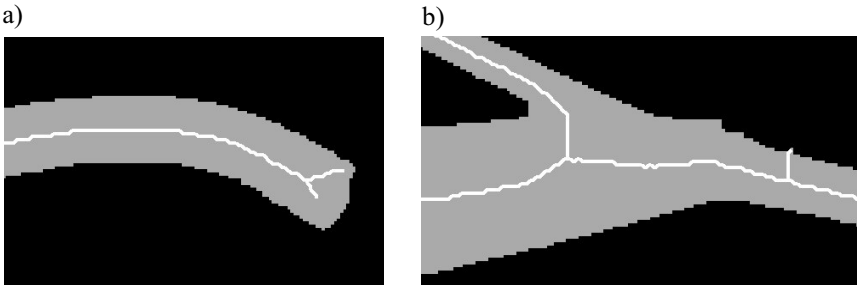


Fig. 7. The example fragments of non smoothed wheat shoot skeleton before pruning with visible false branches: a) the fork at the end of a stem; b) the false branch across a stem

The wheat shoot skeleton can include false branches (paths) going from the midline to the larger ripples of a shoot contour. The branches are much shorter than those mapping main leaf axes because they run across the long and narrow leaf blades of wheat (Fig. 7). To get rid of them, the pruning scheme has been applied to the skeleton graph as in Figure 8 [4, 10].

$$
\begin{aligned}
&\textit{foreach} \quad i \in \{1, K\ , n_V\}, \\
&\quad \textit{if}\ 0 < \textit{length}\left(\mathbf{E}_i\right) < L_{\min}, \\
&\qquad \{\mathbf{E}\} = \{\mathbf{E}\} \setminus \mathbf{E}_i, \\
&\qquad \mathbf{V}(i) = [\ ],
\end{aligned}
$$

Fig. 8. The algorithm of shoot object skeleton pruning, n_V – the number of skeleton end nodes, $\mathbf{V}(i)$ – the array of end nodes, $\mathbf{E}_i$ – the array of i-th end branch.

The operator "[ ]" symbolises deleting a single element from the node array **V** and '\' subtracting from the skeleton graph of all branches **E** shorter than $L_{\min}$. When it is necessary the code illustrated in Figure 8 can be repeated as long as the length condition stays active. By definition and also after pruning the shoot skeleton branches don't touch shoot object contour and therefore must be extrapolated as tangent lines at the end nodes up to the contour. This enables dividing the shoot contour into several distances separated by specific pixels P_i (P_0, ..., P_3 in Fig. 9).

The pixel P_0 labels the shoot base and is assumed to be the rightmost pixel of the contour. On the contrary the shoot tip (P_2 in Fig. 9) is found as the leftmost pixel. The tips of other leaves (P_1, P_3) are pointed by the skeleton end nodes extrapolated to the intersection with the contour line. Let the skeleton nodes and branches after the extrapolation be stored in arrays with the same names as before.

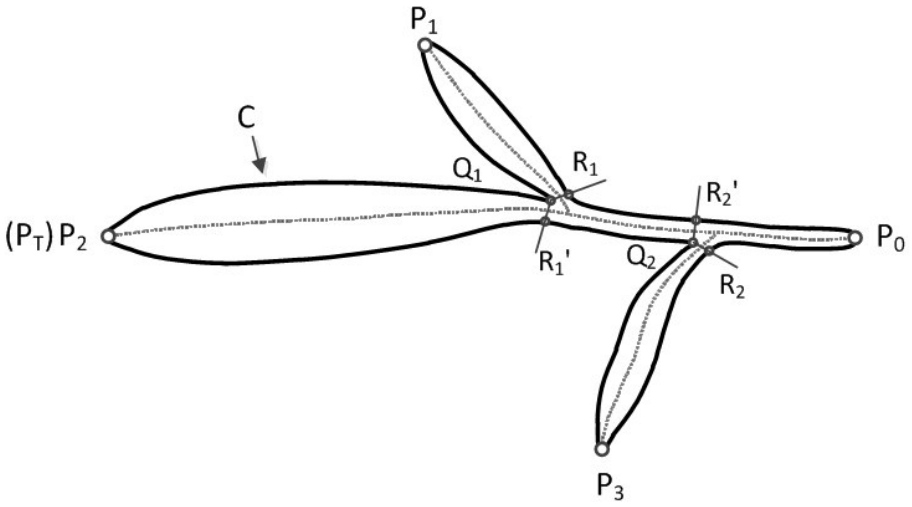


Fig. 9. The topological model of wheat shoot with extrapolated skeleton lines and the characteristic boundary points P_0 – shoot base, P_1, P_2, P_3 – leaf tips, Q_1, Q_3 – leaf branching points, C – shoot contour

Then

$$P_i = \mathbf{E}_i(p_1) = \mathbf{V}(i) \tag{8}$$

where:

$\mathbf{V}$ – the end nodes array of extrapolated skeleton $\mathbf{E}_i$,
$\mathbf{E}_i$ – the extrapolated skeleton branch array beginning with the i-th end node,
p_1 – the first pixel of a branch, $p_1 \in \{P_1, ..., P_M\}$.

The group of leaf branching pixels Q_i (Fig. 9) is required to isolate individual leaves, what is necessary for computing of their areas. Based on the architecture of tested wheat shoots these points can be found as the locally rightmost contour pixels on the contour sections between the pixels P_i, P_{i+1} of two consecutive leaf tips. The contour pixels $C(x, y)$ are traversed from the shoot base P_0 according to the $\mathbf{C}$ array order, but excluding the first and last sections (P_0P_1 and P_3P_0 in Fig. 9). Furthermore the pixels Q_i must be shifted to the right in comparison to the ends of the considered contour distance P_iP_{i+1} and be located far enough from each of them.

$$Q_i = \max_x C(x, y), \quad C(x, y) \in (P_i, P_{i+1}), \quad i \in \{1, \ldots, M-2\} \tag{9}$$

where M – the number of end nodes (their skeleton branches).

A more sophisticated method of Q_i searching can be the detection of a pixel with maximum concavity in the distance P_iP_{i+1} assuming proper averaging level at the convexity computing.

To determine cut-off lines for individual leaves the detected offshoot (branching) pixels Q_i must be associated with the appropriate pixels R_i and R_i' on the opposite sides of shoot leaves. The pixel R_i in Figure 9 represents the location closest to the Q_i in the distance $P_{i-1}P_i$ (or $P_{i+1}P_{i+2}$ for shoot's lower part, below main axis) (Equ. (10)); the pixel R_i' is the next closest location to the Q_i out of $P_{i-1}P_i$ and $P_{i+1}P_{i+2}$ distances, also on the other side of the shoot tip P_T.

$$R_i = C_{jm}(x,y): \begin{cases} jm = \arg\min_j \rho\left(C_j, Q_i\right) \wedge C_j \in P_{i-1}P_i, \quad Q_i \in P_0P_T \\ jm = \arg\min_j \rho\left(C_j, Q_i\right) \wedge C_j \in P_{i+1}P_{i+2}, \quad Q_i \in P_TP_0 \end{cases} \tag{10}$$

where:

ρ – the distance between two pixels,
C_j – the j-th contour point.

Taking above into account leaf separation comes down to the plotting in black straight line distances Q_iR_i, Q_iR_i', $i \in \{1, ..., M-1\}$ inside of the binary image I_B. The incremental line plotting procedure *inclinexy* prepared for this purpose follows line equation through 4-connected neighbours of each pixel (Fig. 10).

$$\begin{aligned} &foreach \quad i \in \{1, K, M-1\}, \\ &\quad I_B' = I_B \wedge inclinexy(I_B, Q_i, R_i, v), \\ &\quad I_B = I_B' \wedge inclinexy(I_B', Q_i, R_i', v). \end{aligned}$$

Fig. 10. The pseudo-code of shoot leaf separation, $v = 0$ – the value associated with cut-off line pixels

5. Digital measurements principles

The proposed algorithm includes the following measurements of shoots geometry:

– the estimation of shoot stem length as well as the lengths of the primary and lower leaves,
– the area calculation of each leaf formerly separated as explained in the previous section.

It is assumed that the leaf lengths correspond to the lengths of end branches of the extrapolated skeleton. The branches are stored as $\mathbf{E}_i$ ($i = 1, ..., n_V$) arrays previously used in Equ. (8). Other lengths are calculated for internal branches of $\{\mathbf{B}_i: i = 1, ..., n_N\}$.The simplest length measurement can be expressed as (Lebowitz) [9]:

$$L_i = \frac{1}{S_F} \sum_{j=2}^{M_i} \left\| P_{j-1}P_j(i) \right\|, \quad \left\| P_{j-1}P_j \right\| = \begin{cases} \sqrt{2} & \text{if } \left|\triangle x_j\right| = 1 \wedge \left|\triangle y_j\right| = 1 \\ 1 & \text{otherwise} \end{cases} \tag{11}$$

where:

L_i – the length of the skeleton branch $\mathbf{B}_i$,
M_i – the number of pixels in the skeleton branch $\mathbf{B}_i$,
$P_{j-1}\,P_j$ – the j-th section between two consecutive pixels along the branch $\mathbf{B}_i$ (Fig. 11) with horizontal and vertical distances Δx_j and Δy_j respectively,
S_F – the *dpi* resolution (in X or Y direction).

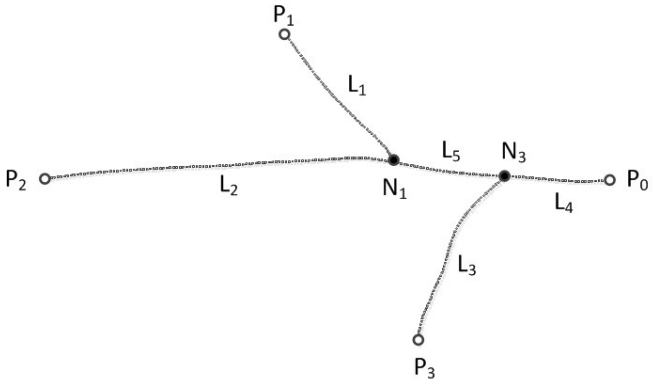


Fig. 11. The skeleton of shoot model from Figure 9. P_0 – shoot base, P_1, P_2, P_3 – leaf tips, L_i – the length of skeleton branch $\mathbf{B}_i$, N_i – internal nodes

This method always gives overestimated lengths because of winding path between consecutive pixels (8 plane directions). To improve length measurement precision every sequence of pixels $\mathbf{B}_i = [(x_j, y_j)]$ representing discrete curve is approximated with cubic splines. For this purpose the function *csaps* from MATLAB's *Spline Toolbox* [16] has been applied in the following way:

$$[x_k \;\; y_k]^T = csaps\left([d_j], [x_j \;\; y_j]^T, \alpha, [d_k]\right)$$
$$j = 1,\ldots,M_i, \;\; k = 1,\ldots,N_i \tag{12}$$

where:

$[x_j, y_j]$ – the input branch pixel array: $\mathbf{B}_i = [(x_j, y_j)]$,
$[d_j]$ – the array of $\mathbf{B}_\mathrm{i}$ pixel distances from the initial pixel P_i,
$[d_k]$ – the array of new distance parameter after smoothing,
α – approximation factor in the smoothing process $\alpha = 0.75$.
$[x_k, y_k]$ – the new branch pixel array after smoothing and subsampling of $\mathbf{B}_i$.

The cubic spline smoothing function can approximate branch pixels in a subpixel grid, denser than the grid of image matrix I_{SK} with an original branch ($N_i \approx 10 \times M_i$). The length measurement in the new grid can be expressed as:

$$L_i = \frac{1}{S_F} \sum_{k=2}^{N_i} \left\| P_{k-1} P_k(i) \right\|, \quad P_{k-1} = (x_{k-1}, y_{k-1}), P_k = (x_k, y_k) \tag{13}$$

where:

L_i – he length of *i*-th shoot branch,
$\|P_{k-1},P_k\|$ – the Euclidean distance (in pixels) of two consecutive branch dense pixels,
S_F – the linear scale factor (in *X* or *Y* direction) set as a *dpi* resolution at a scanning process and then read from JPEG image header,
N_i – the number of dense pixels in the branch $\mathbf{B}_i$.

The computation of shoot partial areas starts from the binary image I_B with leaves isolated by the algorithm contained in Figure 10. The shoot sub-objects are labelled with MATLAB function, which uses the general procedure outlined in [15]:

$$I_L = bwlabel(I_B, N_{CONN}) \tag{14}$$

where:

N_{CONN} – 8-connectivity of object pixels (N_{CONN} = 8),
I_L – the label image for the connected objects.

The calculation of selected properties (e.g. areas) for all labelled objects of the image I_L have been done with the other MATLAB procedure *regionprops* in the following way (Equ. (15)):

$$\mathbf{A}_L = \frac{1}{S_F^2} \left[regionprops(I_L', Area') \right] \tag{15}$$

where:

[ ] – casting from a structure array to the simple array of shoot object areas,
S_F^2 – the constant square of image *xy*-resolution in dpi^2 units.

All inch measures are finally recalculated into cm (or cm^2) units of SI system.

6. Experiments and conclusions

A series of length measurements for the 10 images of wheat shoots have been performed. They were made either manually, using a ruler with millimetre scale, and by one of the automatic methods discussed above. The results are summarized in Table 1. Also the measurements obtained with the simplified method from Equ. (11) have been included for comparison. Both the control group of shoots and a group exposed to selenium were subjects to the measurements.

Table 1
The summary of total shoot length measurement results using different methods

Image number	Methods		
	manual (1) mm	simplified (2) mm	splines (3) mm
1	165.00	169.16	162.19
2	175.00	187.89	176.99
3	183.50	192.85	181.32
4	197.25	203.09	195.98
5	181.25	187.84	182.29
6	193.50	198.93	191.20
7	176.00	185.26	177.36
8	195.50	201.54	192.25
9	200.25	207.41	198.42
10	205.25	218.50	207.07

The error rate of total length measurement method has been defined in relation to the manual method as:

$$e_m = \frac{2\left(\sum_i L_{i,(m)} - \sum_j L_{j,(1)}\right)}{\sum_i L_{i,(m)} + \sum_j L_{j,(1)}} \tag{16}$$

where $L_{i,(m)}$ – the length of i-th branch of shoot skeleton measured with the method m.

The measurement errors of simplified and spline methods are calculated relative to the results of manual method obtained with maximum accuracy of ¼ mm because of ruler scale. The error rates have been computed according to the formula in Equ. (16), instead of using the average relative error of the skeleton branch length measurement. This is due to the fact, that up to now the manual length measurements were carried out for the shoots as a whole and the measurement precision of individual leaves has not been considered. Additionally manual methods can sometimes detect small leaves invisible in the shoot image, so the two different indexes i and j of length sums have been used in the numerator of Equ. (16). The difference of summary lengths is related to the average of these lengths, because the deviation of measurements between manual and automatic method is to be measured.

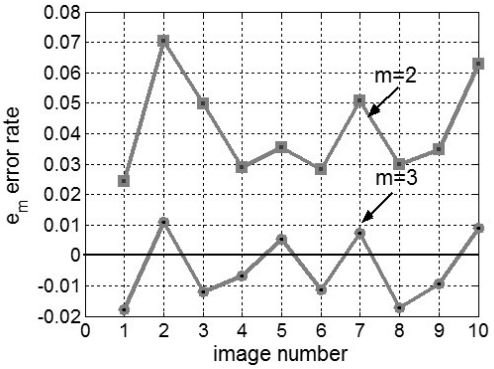


Fig. 12. The error rate plots for the measurements of total shoot lengths with the methods 2 (m = 2) and 3 (m = 3) according to the Eqs. (11) and (13).

The error rates of the simplified image method (m = 2 in Fig. 12) are always greater, than these obtained manually using the ruler, on average 4.2%. When tracking 8-adjacent image pixels along skeleton branches, the obtained sum of neighbour pixel distances is usually greater than real branch length.

The measurement results presented in the paper have been used by biologists from the Department of Plant Physiology and Biochemistry of University of Łódź for further comparison of wheat growth in different conditions.

This drawback is avoided by sub-pixel following of the branches with the help of cubic spline approximation (Eqs. (12) and (13)). Therefore the measurements with the spline method oscillate around the manual results with the Equ. (16) errors in the range of [–1%,+2%].

The shoot area measurements were taken only for the comparison of shoot control groups with those exposed to selenium. So the example results listed in Table 2 were not related to manual measurements.

The execution times of the measurement algorithm for ten example images are shown in Table 3. The times mainly depend on the sizes of images, which are cut as windows around shoot objects from the whole A4 format at 300 *dpi* of horizontal resolution. The average execution time in Table 3 is equal to 2.7 s and can be fully acceptable for the considered type of measurements.

The method presented in the paper has two important parameters:

1) $L_{\min}$ – pruning distance in the code in Figure 8 as the minimum number of pixels in a shoot skeleton branch to be cut-off,
2) N – number of harmonics preserved at EFD contour smoothing.

Both parameters are available as controls in the main dialog window of the method's application to adjust them to different shoot types. The parameters are set intuitively, only once for the given class of shoot objects and after the proper setting need not be change at any measurement.

Table 2
The summary of shoot main leaf area measurements for the control and selenium treated groups of wheat seedlings

Image number	Shoot's primary leaf areas		
	Control group mm^2	Group (1) selenium 15 µM mm^2	Group (2) selenium 2.5 µM mm^2
1	483.68	480.30	346.50
2	545.65	484.10	637.86
3	484.62	608.75	447.47
4	553.92	530.75	509.49
5	518.63	532.88	520.05
6	450.72	535.45	446.11
7	413.30	445.59	434.82
8	546.36	567.88	563.82
9	410.44	560.21	563.35
10	679.85	559.77	505.15
average	508.7	530.6	497.5

Table 3
The spline measurement execution times for the example images

Image number	Execution time [s]	Image resolution [px]
1	2.953	563×2030
2	2.589	650×1904
3	3.127	750×2017
4	2.082	463×2005
5	2.092	450×2042
6	2.407	463×2042
7	2.622	613×1942
8	2.637	575×2080
9	3.465	825×1929
10	2.804	625×2030

Acknowledgements

The research was a part of investigations carried out with collaboration of the Department of Plant Physiology and Biochemistry of the University of Lodz. The authors wish to acknowledge dr Ewa Gajewska and dr Marzena Wielanek for sharing the results of their experiments on plants, especially image data and precious comments, which helped us develop the ideas.

References

[1] Cheng H.D., Jiang X.H., Sun Y., Wang J., *Color image segmentation: advances and prospects*. Pattern Recognition, vol. 34, 2001, 2259–2281.

[2] Gocławski J., Sekulska-Nalewajko J., Gajewska E., Wielanek M., *Automatyczny pomiar długości korzeni siewek pszenicy z hodowli hydroponicznej przy wykorzystaniu metod przetwarzania i analizy obrazów*. Automatyka (półrocznik AGH), t. 13, z. 3, 2009, 831–847.

[3] Gocławski J., Sekulska-Nalewajko J., Gajewska E., Wielanek M., *An automatic segmentation method for scanned images of wheat root systems with dark discolourations*. International Journal of Applied Mathematics and Computer Science, vol. 19, No 4, 2009, 679–689.

[4] Gonzalez R.C., Woods R.E., *Digital image processing*, 3rd ed. Prentice Hall, 2007.

[5] Gonzalez E.R., Woods R.E., Eddins S.L., *Digital Image Processing Using MATLAB*. Prentice Hall, Upper Saddle River, NJ, 2004.

[6] Hasthorpe J., Mount N., *The generation of river channel skeletons from binary images using raster thinning algorithms.* URL http://ncg.nuim.ie/gisruk/materials/ proceedings/PDF/P7.pdf, 2007.

[7] Kuhl F.P., Giardina C.R., *Elliptic Fourier features of a closed contour.* Computer Graphics and Image Processing, vol. 18, 1982, 236–258.

[8] Lam L., Lee S., Suen Y., *Thinning Methodologies – A Comprehensive Survey.* IEEE Transactions on Pattern Analysis and Machine Inteligence, vol. 14, 1992, 869–885.

[9] Lebowitz R.J., *Digital image analysis measurement of root length and diameter.* Environmental and Experimental Botany, vol. 28, 1988, 267–273.

[10] Malina W., Smiatacz M., *Metody cyfrowego przetwarzania obrazów.* Akademicka Oficyna Wydawnicza EXIT, Warszawa 2005.

[11] Neto C.J., Meyer G.E., Jones D.D., Samal A.K., *Plant species identification using Elliptic Fourier leaf shape analysis.* Computers and Electronics in Agriculture, vol. 50, 2006, 121–134.

[12] Otsu N., *A Threshold Selection Method from Gray-Level Histograms*. IEEE Transactions on Systems, Man, and Cybernetics, vol. 9, No. 1, 1979, 62–66.

[13] Serra J., *Introduction to Mathematical Morphology.* Computer Vision Graphics and Image Processing, vol. 35, 1986, 283–305.

[14] The Mathworks Inc.: *MEX-files Guide*. URL http://www.mathworks.com/ support/tech-notes/ 1600/1605.html.

[15] The Mathworks Inc.: *Image processing toolbox user's guide.* URL http://www.mathworks.com/ help/toolbox/images/.

[16] The Mathworks Inc: *Curve Fitting Toolbox.* URL http://www.mathworks.com/ help/toolbox/ curvefit/.