Automatic detection of cereal rows by means of pattern recognition techniques

Henri Tenhunen[a], Tapio Pahikkala[a], Olli Nevalainen[a], Jukka Teuhola[a], Heta Mattila[b], Esa Tyystjärvi[b,*]

[a] *Department of Future Technologies, Computer Science, University of Turku, Finland*
[b] *Department of Biochemistry, Molecular Plant Biology, University of Turku, Finland*

Abstract

Automatic locating of weeds from fields is an active research topic in precision agriculture. A reliable and practical plant identification technique would enable the reduction of herbicide amounts and lowering of production costs, along with reducing the damage to the ecosystem. When the seeds have been sown row-wise, most weeds may be located between the sowing rows. The present work describes a clustering-based method for recognition of plantlet rows from a set of aerial photographs, taken by a drone flying at approximately ten meters. The algorithm includes three phases: segmentation of green objects in the view, feature extraction, and clustering of plants into individual rows. Segmentation separates the plants from the background. The main feature to be extracted is the center of gravity of each plant segment. A tentative clustering is obtained piecewise by applying the 2D Fourier transform to image blocks to get information about the direction and the distance between the rows. The precise sowing line position is finally derived by principal component analysis. The method was able to find the rows from a set of photographs of size 1452×969 pixels approximately in 0.11 seconds, with the accuracy of 94 per cent.

Keywords: computer vision, pattern recognition, principal component analysis, Fourier transform, precision agriculture

*Corresponding author, esatyy@utu.fi

1. Introduction

Machine vision, based on imaging by satellites, aeroplanes or drones, is a central research field in precision agriculture. Due to getting cheaper and more versatile, drones have quickly gained popularity in precision agriculture. Images taken from low-flying drones are in many cases precise enough for inspection of individual plants – something not reachable by the means of satellites (Maes and Steppe, 2018).

There have been prior projects for estimating the number, height, and locations of individual plants (see for example Søgaard and Olsen 2003; Tang and Tian 2005; Bakker et al. 2008a; Bakker et al. 2008b; Jiang et al. 2015). The health of plants, as well as the existence and location of weeds are also important issues to be observed (Woebbecke et al., 1995; Southall et al., 2002; Burgos-Artizzu et al., 2011). Traditional farm machines or more modern robots (usually on wheels) can be used to locally spray nutrients on small areas, or herbicides on the detected weeds. By local application of herbicides, reductions between 4 to 94 per cent in the herbicide amounts have been obtained in field experiments (de Castro et al., 2012, 2013).

The aim of the present study was to design novel methods for early detection of monocot weeds between cereal rows. Monocot weeds cannot easily be distinguished from the cereals by form or color, and their identification is possible only at a later stage when it may be too late for weed removal operations. We are particularly interested in common wild oat (*Avena fatua L*), a weed widely infesting cereals in Europe.

Early detection of monocot weeds can be based on the fact that cereals are sown in regular rows whereas most weeds would not grow within the rows. For weed detection, automatic classification of each plant in a field photograph to either a plantlet growing in a row or a plant growing in-between the rows will help to detect most weeds and to identify weed-infested areas. In the present study, we took the first step toward identifying plants growing in rows and between rows by developing a method for locating the rows in drone photographs

taken from a weedless field.

2. Previous research

Research on methods for the detection of crop rows has focused on guiding of a tractor or an agricultural robot along the crop rows. For this reason, images used for row detection have typically been taken from an altitude of maximally two meters and show only few crop rows that run from the bottom of the picture to the top with a strong perspective (Marchant and Brivot 1995; Kise et al. 2005; Bossu et al. 2009; Anter et al. 2019; Jiang et al. 2015; Jiang et al. 2016; Zhang et al. 2018). Furthermore, the tested agricultural settings have been mostly limited to conditions in which such robotic devices are feasible, and therefore relatively large plants with a clear row structure like cauliflower (Marchant and Brivot, 1995), maize (Zhang et al., 2018), soybean (Kise et al., 2005), sugar beet, canola (Winterhalter et al., 2018) and potato (Vidović et al., 2016), or plants with dense rows (Jiang et al., 2016) or large row spacing like in paddy rice (Choi et al., 2015) have been tested. The methods are designed to function in real-time, and therefore low-resolution pictures are often preferred, from 3000x2250 (García-Santillín et al., 2018) down to 320x240 resolution (Vidović et al., 2016). The most common camera system has been a standard RGB camera, although also NIR cameras (Kise et al. 2005; Choi et al. 2015) and stereo vision or laser scanning (Winterhalter et al., 2018) have been used.

Several computational methods have been applied to row detection. All these methods need knowledge about the positions of the plant pixels. In RGB photographs, the plant/soil discrimination is usually done by calculating the greenness of a pixel by a vegetation index (e.g. Woebbecke et al. 1995, Motohka et al. 2010, Burgos-Artizzu et al. 2011, and Jiang et al. 2015) and refined with thresholding with Otsu's method (Otsu, 1975) or by k-means clustering (Han et al., 2004). Søgaard and Olsen (2003) avoid the segmentation step in their row recognizer. They convert the colour image into gray-scale by a linear transform emphasizing the green channel. After plant/soil separation, methods detecting

individual plants often calculate a single point defining the position of the plant; methods used for this task depend strongly on the plant species. For example, the horizontal center of an individual paddy rice plant is a well-defined concept when the plant is visible in a low-altitude picture taken from an inclined position (Choi et al., 2015).

The simplest methods designed for the purpose of guiding a vehicle are based on the idea that the distribution of plant pixels in a horizontal slice of the photograph concentrates on the rows (Reid and Searcy 1987; Olsen 1995; Søgaard and Olsen 2003). These methods work best when the areas between plant rows are homogeneous, i.e. free of weeds. In addition, rows with sparsely growing plants will cause difficulties for these methods. The crop row detection methods have been refined with the idea that in a perspective picture, the rows point to a common vanishing point (Pla et al. 1997; Jiang et al. 2016). These methods function only for pictures in which the rows run approximately from bottom to top but they have the advantage of not being dependent on any specific method of defining the center of an individual plant.

A more advanced method that has extensively been applied to row detection is Hough transform that uses a voting scheme to detect line segments in pictures (Marchant and Brivot 1995; Bakker et al. 2008a; Choi et al. 2015; Jiang et al. 2016; García-Santillín et al. 2018). The Hough transform is generally quite slow, which restricts its utilization in real-time detection. The computing time can be decreased by first defining the plant center (Choi et al., 2015). The Hough transform can detect rows of any direction in the image, but if the rows have gaps or there are weeds in the picture, the Hough transform may detect additional lines that are not parallel with the crop rows (Marchant and Brivot 1995; Choi et al. 2015). Therefore the results need to be refined after the initial Hough transform. Application of the wavelet transform for row detection has given relatively long computation times (Bossu et al., 2009).

The third method commonly used for row detection is linear regression (Billingsley and Schoenfisch 1997; Søgaard and Olsen 2003; Han et al. 2004; Jiang et al. 2015). In this method, the picture can be divided to windows

around each candidate point, and candidate crop rows are obtained by calculating linear regression of the candidate points inside each separate windows. The validity of each regression line is then assessed e.g. by calculating the moment of inertia of the pixels of the window with regard to the candidate row (Billingsley and Schoenfisch 1997; Jiang et al. 2015). García-Santillín et al. (2018) applied curvilinear regression models in combination with the Hough transform for detection of curved rows. Linear regression does not depend on the row directions but would lead to a very large number of candidate rows in a picture taken from above of a field, with large gaps in rows.

Recent research on row detection methods has led to suggestions for global fitting methods that seek to avoid errors caused by gaps, weeds, plant morphology and local or global curvature of rows. These methods include the global energy minimization method of Vidović et al. (2016) that looks for regular patterns and the pattern Hough transform method for simultaneous detection of all rows and row spacing (Winterhalter et al., 2018). The use of 3D imaging (Winterhalter et al., 2018) is another potential future method, as the laser scanning technique develops rapidly.

In the present study, the photographs were taken from a moderate altitude from above the field. Therefore many of the previous methods for row detection are not directly applicable to our case. In the method of Jiang et al. (2015), the colour image is initially segmented by transforming it first into a gray-scale image on the basis of a vegetation index (Burgos-Artizzu et al., 2011). The gray-scale image is then binarized, by using a threshold given by the method of Otsu (1975). In the second phase, the center points of the seed rows will be determined by solving an optimization problem that assumes a constant distance between adjacent rows. The solution produces both real and erroneous center points. The latter ones will be filtered using a classification algorithm. The seed rows are finally localized by the least squares method. The approach is promising, and we therefore started our work by applying it to our image data. It turned out that our implementation of the method, with certain carefully selected parameter values and images of certain particular directions of seed

rows, produced excellent results (Figs. 1a and 1b).

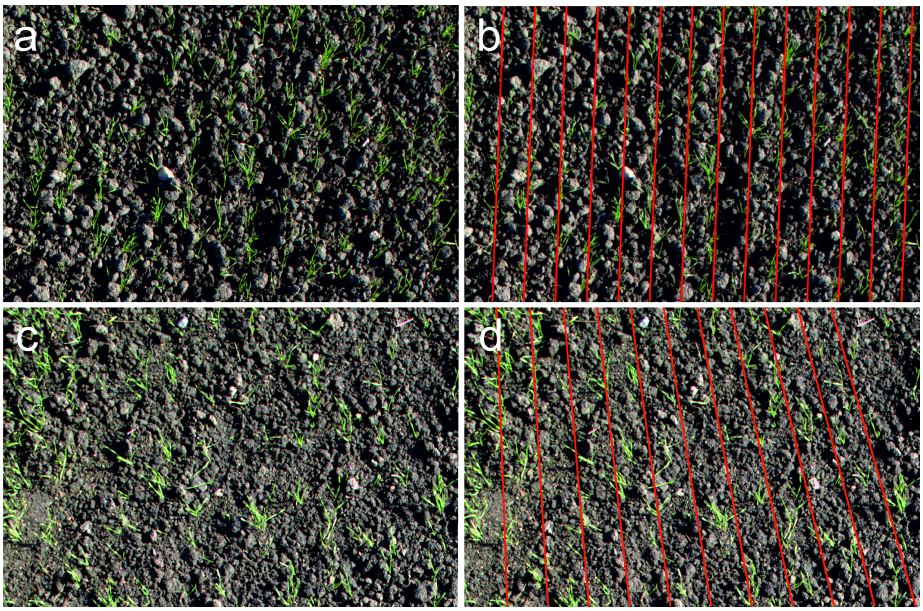

Figure 1: Recognition of seed rows by the method of (Jiang et al., 2015)

As a drawback, the method of Jiang et al. (2015) cannot be directly applied if the seed rows are horizontal. The method also tends to fail when the initial center points of rows should be searched in regions with sparse occurrences of plants (Figs. 1c and 1d). The reason for this is that the scanning of the first row is crucial for the success of the whole recognition process, as the remaining center points will be found in relation to this line.

Another approach was to apply Hough transform to the binarized image data. As mentioned before, large gaps in the rows cause the method to detect additional lines that do not match the rows. One way to overcome this problem is to restrict the parameter space by searching only lines that are in the same direction as the sowing rows and within the allowed distance from each other. The directional angle θ of the rows can be estimated automatically (to be described in Section 4). Other parameters are predefined, such as minimum line length, a distance between adjacent rows and the minimum value to be

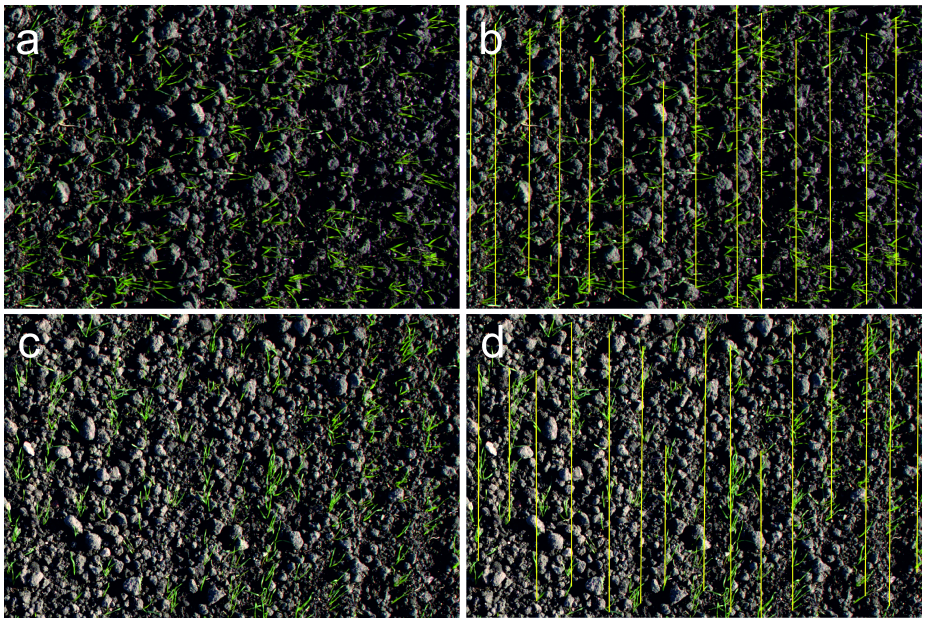

Figure 2: Recognition of seed rows by Hough transform combined with automatic detection of direction of the rows.

considered as a peak in the the accumulator space.

By tuning these parameters, the method is capable of producing good results (Figs 2a and 2b). However, rows with sparsely growing plants cannot be completely detected (Figs 2c and 2d) and the performance is sensitive to seven handcrafted parameters, which limits its application.

The row recognition algorithm used in the present study is based on clustering of plants into rows. We took photographs of a rye field when the shoots of the cereal were about 2-7 centimeters high and the overlap between individual rye plantlets was modest. The algorithm includes two main steps: segmentation of plant areas from the background (soil, stones, shadows, etc.), and their assignment to row clusters. We suggest two alternative algorithms for plant segmentation; *k-means clustering*, and a method that utilizes a transform of the colour channel values, combined with *Otsu's method* (Otsu, 1975). The k-means method turned out to be accurate, but unfortunately rather slow. An alterna-

tive *color channel transformation method* was significantly faster, but required more image processing steps.

3. Problem setting

We are given a set of colour photographs taken by a drone which scans a crop field at the flying height of approximately 10 meters. The task is to design and implement a method which

(a) separates the individual plants from the background,

(b) localizes the areas of the image containing the rows of plants,

(c) allows gaps in the rows, to a certain extent.

Unlike the works of Tang and Tian (2005), Southall et al. (2002), Bakker et al. (2008a), and Jiang et al. (2015), the demand for real-time operation of the technique is relaxed. Note that the mentioned earlier approaches are for applications where the photos are supposed to be taken from a moving tractor, coupled with spraying the herbisides, whereas in our case a drone is doing only the data collection for aiding later operation of a weed removal device.

It is assumed that the rows can be modelled as straight lines of the form

$$(x, y) = (\alpha + \beta y, y)\,, \tag{1}$$

where β is the slope of the line. The equation allows the description of both inclinated lines and those parallel to the y axis. For practical purposes, it is advisable that the drone does not significantly change its orientation with respect to the underlying surface, and the shooting direction of photos is vertical, i.e. the y-axis of the camera approximately parallels with the moving direction of the drone. This is not mandatory for our row detection method, but helps in systematic covering of the field. The need for perspective correction is avoided by dividing the images into sufficiently small windows, within which the row detection is applied separately. In practice, the seed rows consist of groups of parallel rows, where each group corresponds to an array of feeders in the sowing

machine. Within groups, the distances between adjacent rows are more or less the same, but between groups, they may vary, due to the possible overlap or gaps between the driving lanes of the sowing machine. In this paper we concentrate on the intra-group modelling, but discuss the options for handling the general case in the conclusions.

On a coarse level the proposed row recognizer consists of three main phases; segmentation, feature extraction, and clustering. In segmentation, the plants are separated from the background. After this, they are clustered into disjoint rows, which are finally modelled by the least squares method.

4. Row recognizer

The row recognizer accepts as its input a set of colour photographs, representing rectangular subareas of a cereal field, taken by a drone from above. The task of the recognizer is to produce a set of detected sowing rows for each image. Each row is represented by a sequence of line segments that together constitute the row and visualized by overlaying the lines on the original image. Another visualization to be used is to mark the adjacent rows with different colours. Joining the neighbouring, partially overlapping images together is a different type of problem and not discussed here.

4.1. Segmentation

Recognition of plants from raw data may be problematic without preprocessing. Young plants, being our main interest here, are small and can be difficult to separate from a dominating background, especially on a sunny weather (see Fig. 3a). In this figure, the actual sowing rows are straight but the sowing machine was turned at the site after sowing, and the back rake of the machine made the curved furrows.

The segmentation of the plant objects is attempted here by two alternative algorithms: k-means clustering of image elements (Lloyd, 1982), and a *colour channel transform* combined with Otsu's method. In order to make the work more self-contained, we will briefly recall their main ideas below.

k-means clustering of image elements

In general, the k-means clustering algorithm obtains a data set X of n objects $x_j (j = 1, \ldots, n)$, where $x_j = (x_{j1}, x_{j2}, \ldots, x_{jr})$, for a fixed dimension $r \geq 1$. The aim of the clustering is to search, for a given k, a partition $C^* = (C_1^*, C_2^*, \ldots, C_k^*)$ such that

$$C^* = \underset{C}{\operatorname{argmin}} \sum_{i=1}^{k} \sum_{x \in C_i} ||x - \mu_i||^2 \tag{2}$$

where C stands for all possible partitions of X, and μ_i the center point of the objects $x \in C_i$. The '||' operator stands for the vector norm which is here assumed to be simply the Euclidean distance (the L_2 norm). In other words, one wants to find a set of k disjoint clusters covering all the objects in X such that the sum of intra-cluster distances from the cluster center points is minimal.

We notice that k-means clustering can be applied in a natural way to a colour image of $n = N \times M$ pixels in the *CIE colour space*. The data set $X = [x_1, x_2, \ldots x_n]^T$, where each $x_i \in X$, is a two-dimensional vector containing the values of colour channels a_i^* and b_i^* of pixel i. (Intensity channel L^* is omitted here, due to changing lighting conditions.) When choosing $k = 3$, the clusters of C^* stand for C_R, C_G, and C_B (closest to red, green, and blue, respectively). In row recognition, cluster C_G should stand for plant pixels (see Fig. 3b).

The method gave good results for our data, but as a drawback the execution times are long because of the inherent NP-hardness of the problem (see Appendix, Table 3).

Gray-scale transform and binarization with Otsu's method

It is typical of the previous plant row recognizers that individual plants have been found by some kind of transformation of colours into gray-scale values. This can be done by enhancing the values of the green channel, and at the same time reducing those of blue and red (see Jiang et al. 2015, Woebbecke et al. 1995, and Motohka et al. 2010). In the present work we apply the vegetation index I_H discussed by Burgos-Artizzu et al. (2011):

$$I_H(x, y) = I_g(x, y) \times 1.262 - I_r(x, y) \times 0.884 - I_b(x, y) \times 0.311\,, \tag{3}$$

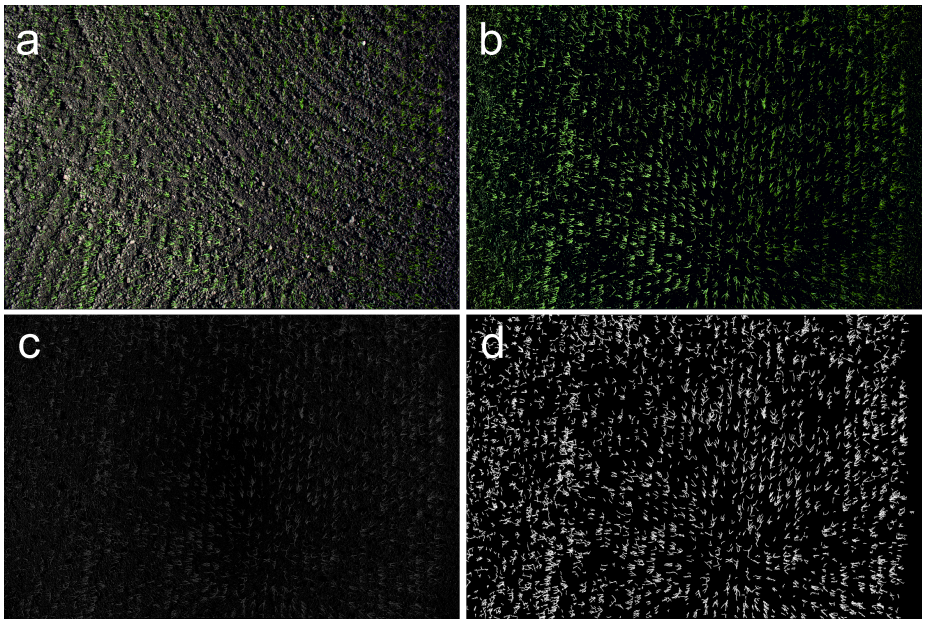

Figure 3: a) A sample colour photo of young plants, without preprocessing. b) The effect of k-means clustering to Fig. 3a. The pixels of the C_G-cluster are shown. c) The effect of the vegetation index transform for Fig. 3a, converted to gray-scale. d) The result of preprocessing Fig. 3a by using the vegetation index, thresholding by Otsu's method, and filtering off small connected components.

where (x, y) stands for a pixel in an $M \times N$ image I (see Fig. 3c, for an example). The transform clearly works in $O(M \times N)$ time.

A gray-scale image I_H can be transformed into a binary image by the mapping (Gonzalez and Woods 2002)

$$I_B(x,y) = \begin{cases} 1, & \text{if } I_H(x,y) > t \\ 0, & \text{if } I_H(x,y) \leq t \end{cases} \tag{4}$$

Threshold t is selected to minimize the weighted sum of the internal variances of the pixel values in the 0-pixel and 1-pixel groups Otsu (1975):

$$\sigma^2_\omega(t) = \omega_0(t)\sigma^2_0(t) + \omega_1(t)\sigma^2_1(t)\,, \tag{5}$$

where $\sigma^2_0(t)$ is the variance of gray-scale values less than or equal to the threshold t; these pixels are thus changed to 0-bits. Similarly $\sigma^2_1(t)$ is the variance of the

gray-scale values greater than t. Further, the term $\omega_0(t)$ is the total number of 0-bits, and $\omega_1(t)$ is the total number of 1-bits, for a given t.

In the last step we remove small binary fragmentary objects that can be considered noise. This is done by expressing the 0/1 image as an undirected graph G of $M \times N$ nodes $V = \{(x, y)\}$ where the nodes stand for the pixels of the image $I_B(x, y)$. There is an arc in G between node (x, y) and each of its eight neighbours $((x-1, y-1), (x, y-1), \ldots (x+1, y+1))$ if both (x, y) and its neighbour contain a 1-bit (Bai et al. 2012). Then we define the connected components of G by a fast algorithm and filter out those components which include less than a given limit t_{cc} of nodes. This process changes $I_B(x, y)$ into a filtered image $I'_B(x, y)$. The value of t_{cc} naturally depends on the application. In our data, manual inspection of a sample gave the average plant size of somewhat over 160 pixels. In Fig. 3d, the connected components with less than 50 pixels have been filtered out. The value of parameter t_{cc} should be derived from the training data of the plant row recognition system.

4.2. Feature extraction

In the previous section the plants were separated from the background. Now we introduce features which will be used for automatic determining of the number of sowing rows, and mapping of plants to their rows. For doing this, consider a connected component i. We determine the point of gravity $(\bar{x}_i, \bar{y}_i)$ of i, and the angle θ of the straight line approximating the sowing row. The angle will be found from the largest coefficients of the *discrete Fourier transform* of the binary image I'_B, from which the smallest factors have been removed (see e.g. Cormen et al. 2001, Chapter 30). The feature to be determined is simply the x-coordinate of the component's point of gravity in a transformed space, obtained as follows.

$$\bar{x}'_i = [(\bar{x}_i - N/2)(\bar{y}_i - M/2)] \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} + N/2 \qquad (6)$$

Center of gravity

Consider a connected component R_i in the filtered binary image I'_B. The geometric center of gravity of R_i is then $(\bar{x}_i, \bar{y}_i)$

$$\bar{x}_i = \sum_{(x,y)\in R_i} (x/|R_i|) \text{ and } \bar{y}_i = \sum_{(x,y)\in R_i} (y/|R_i|)\,, \tag{7}$$

where $|R_i|$ is the number of pixels in R_i.

Directional angle θ of the rows

The Fourier transform can be used to convert a signal into a sum of sine and cosine functions. The transform is useful in image applications because the low frequency coefficients of the transformed signal correspond to the main features of the original signal in spatial space.

A binary image I'_B can be transformed by the two-dimensional discrete Fourier transform to the form

$$I_F(k,l) = \sum_{m=1}^{M}\sum_{n=1}^{N} I'_B(m,n) e^{-j2\pi km/M} e^{-j2\pi ln/N}\,, \tag{8}$$

where $j = \sqrt{-1}, k = 1, 2, \ldots, M$, and $l = 1, 2, \ldots, N$. The result can be interpreted as a matrix (or image in the frequency space) where the DC-coefficient (called 0-frequency) with indexes $(1, 1)$ is in the left-upper corner. For the needs of interpreting the direction of the sowing rows, the frequency space is shifted so that the 0-frequency is in the middle of the coordinate space, as shown in Fig. 4. This rearranged transformed "image" is called the I_C image of the original spatial binary image.

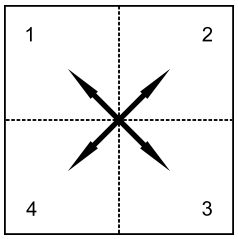

Figure 4: Centering the 0-frequency by interchanging the quadrants $1 \leftrightarrow 3$ and $2 \leftrightarrow 4$.

Fig. 5 shows a 3-dimensional I_C view of a typical image in our application. The coefficients for low-frequency functions are in the middle (marked by yellow) in this view. The high peaks close to the 0-frequency peak contain information about the direction of the regularly repeating patterns (here for the parallel, equi-distant sowing rows) in the original image. Note here that the "peaks" in I_C consist of several high coefficient values around the absolute maximum.

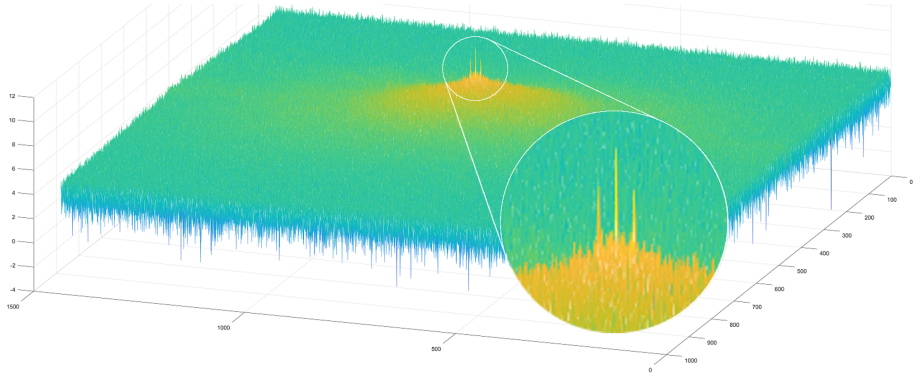

Figure 5: 3D view of the Fourier transform.

For automatic extraction of the peaks, image I_C is filtered twice. First, the 0-frequency peak is removed by setting the 6×6 square in the middle of I_C to zeroes (experimental choice). This gives us a coefficient matrix $I_{\neg DC}$. Secondly, we want to find a small number of the highest remaining peaks from $I_{\neg DC}$. It turns out that when the plant rows are not extremely sparse, two peaks suffice (actually one because the transform is symmetric with respect to the origin). Nevertheless, the algorithm works well for other settings, too, up to 10 peaks. The search for the highest peaks will be done by thresholding $I_{\neg DC}$ with different increasing threshold values, denoted p, and iteratively searching for the largest p-value which yields the wanted number of peaks[1], which form the basis for a good estimate of the required rotation angle. In fact, a regression line through the peaks should be perpendicular to the direction of rows. The

[1]Plain sorting of the elements of I_F would be the other option, but thresholding is easier when looking for connected clusters of low-frequency values.

thresholding produces a binary matrix $I_T^*(x, y)$ of size $M \times N$, where the peaks appear as 1-bits, others being zeroes.

Josso et al. (2005) have shown how the *principal component analysis* (*PCA*) can be applied to estimate the texture orientation of an image, on the basis of high peaks of the Fourier transform. Here we apply their method for finding the angle of the sowing rows, with respect to the basic direction determined by the camera position.

For the needs of PCA, let $X_T = (x_1, x_2, \ldots, x_q)$ and $Y_T = (y_1, y_2, \ldots, y_q)$ give the (x, y)-coordinates of the 1-pixels in the thresholded binary image I_T. Calculate the *covariance matrix*

$$\Sigma = \begin{bmatrix} \sigma(X_T, X_T) & \sigma(X_T, Y_T) \\ \sigma(Y_T, X_T) & \sigma(Y_T, Y_T) \end{bmatrix} \tag{9}$$

where

$$\sigma(X_T, X_T) = E[(X_T - E(X_T))(X_T - E(X_T))] = \frac{1}{q-1} \sum_{i=1}^{q} (x_i - E(X_T))^2$$

$$\sigma(Y_T, Y_T) = E[(Y_T - E(Y_T))(Y_T - E(Y_T))] = \frac{1}{q-1} \sum_{i=1}^{q} (x_i - E(Y_T))^2$$

$$\sigma(X_T, Y_T) = \sigma(Y_T, X_T) = E[(X_T - E(X_T))(Y_T - E(Y_T))]$$

The largest eigenvectors of the covariance matrix Σ point to the directions of the (x, y)-space containing the biggest amount of variance. The eigenvectors $\bar{v} \in R^2$ ($\bar{v} \neq 0$) and eigenvalues λ fulfill the equality $\sum \bar{v} = \lambda \bar{v}$, and the principal component of this vector equation is

$$v_1 = \text{argmax}_{||v||=1} \{\bar{v}^T \Sigma^T \Sigma \bar{v}\}, \tag{10}$$

where $v_1 = (v_y, v_x)$ is the eigenvector corresponding to the largest eigenvalue λ_1. Since we are dealing with frequency peaks of the Fourier transform, the eigenvector is perpendicular to the plant rows. The wanted general angle of the sowing rows is then

$$\theta = -(\text{atan2}(v_y, v_x)), \tag{11}$$

where atan2(x, y) is the angle between the positive x-axis and the ray to the point (x, y).

4.3. Unsupervised learning step

After the feature extraction, the next step is to map plants to their rows. The purpose is to split a data set $\bar{X}'$ containing n plant positions $\bar{x}'$ into clusters $K = \{K_1, K_2, \ldots, K_k\}$ where k is the number of rows. Here we use *k-medoids*, a classical partitioning method which seeks for k "representative entities" that minimize the overall dissimilarity of every entity of the data set to the nearest representative entity (Kaufman and Rousseeuw, 1987). We define a cluster as the set of entities which have been assigned to the same representative entity. Clusters are derived from decision variables y_i and z_{ij} with the following optimization model:

$$\text{minimize} \sum_{i=1}^{n} \sum_{j=1}^{n} \|\bar{x}'_i - \bar{x}'_j\|^2 z_{ij} \tag{12}$$

subject to

$$\sum_{j=1}^{n} z_{ij} = 1, j = 1, 2, ..., n \tag{13}$$

$$z_{ij} \leq y_i, i, j = 1, 2, ..., n \tag{14}$$

$$\sum_{i=1}^{n} y_i = k, k = \text{number of clusters} \tag{15}$$

$$y_i, z_{ij} \in \{0, 1\}, i, j = 1, 2, ..., n \tag{16}$$

where y_i is equal to one if and only if $\bar{x}'_i$ is assigned as a representative entity and z_{ij} is equal to one if and only if $\bar{x}'_j$ is selected to the cluster of which $\bar{x}'_i$ is the representative entity (Kaufman and Rousseeuw, 1987). K-medoids requires the number of clusters to be pre-specified. Therefore, the next phase is to determine the number of sowing rows in a an image.

Number of rows

Locations of the sowing rows are relocated in the spatial image by a brute force technique where the binary image $I_B(x, y)$ is rotated counter-clockwise by

the angle $\pi - \theta$, so that the rows turn vertical with respect to the x-coordinate axis. Denote the rotated image by I_R. All pixel positions in the x-axis from x_1 to x_n are then considered, and the number of corresponding 1-bits in each column is calculated:

$$S(x_i) = \sum_{y=1}^{M} I_R(x_i, y) \tag{17}$$

The idea is to find the x-ranges that contain a lot of 1-pixels, and in this way to locate the sowing rows. While doing this, one may filter out some fragmentary 1-pixel objects by the median filter

$$S_{med}(x_i) = \text{med}(S(x_i - \omega), S(x_i - \omega + 1), \ldots, S(x_i + \omega)), \tag{18}$$

where ω is a user-defined parameter (see Huang et al. 1979). In our application, values of ω from the range 1 to 15 worked well. Suppose that x^* is a local maximum point for $S(x_i)$. It is then accepted as a center point of a sowing row if there is no larger median within any interval $(x-\omega, \ldots, x+\omega)$ for $x \in [x-\epsilon, x+\epsilon]$, and if $S(x^*)$ is not smaller than a threshold ζ:

$$(S_{med}(x^*) \geq S_{med}(x)) \text{ and } (S(x^*) \geq \zeta) \tag{19}$$

for $|x-x^*| \leq \epsilon$. Parameter ζ is a predefined constant, representing the minimum accepted number of green pixels in a pixel column. The experimentally obtained values $\zeta = 20$ and $\epsilon = 75$ were applied. The number k of sowing rows is now the number of local maximum points (x^*) found in the above process, i.e. if $X^* = \{x_1^*, x_2^*, \ldots, x_k^*\}$, then $k = |X^*|$, see Fig. 6, for an example.

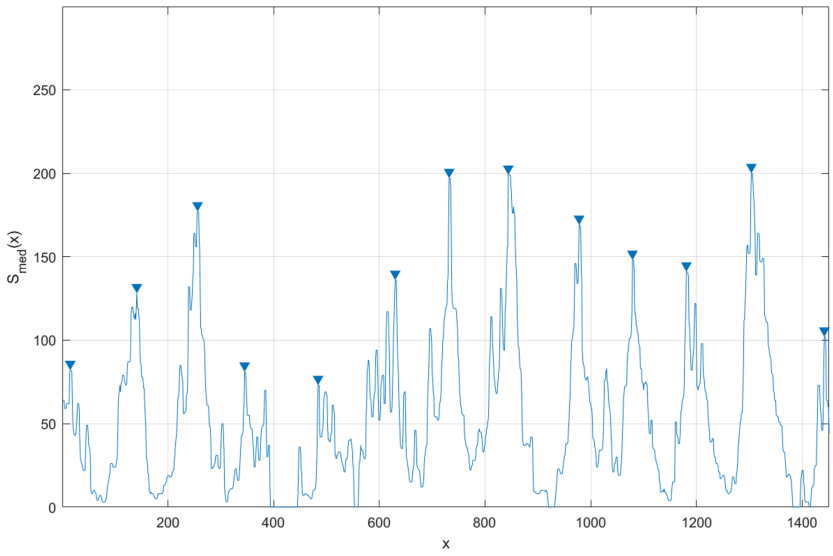

Figure 6: Finding frequency peaks after rotating the sowing rows to vertical.

Linear regression

The result of assigning the points of gravity to the approximated rows makes the estimation of the sowing rows possible. We therefore estimate the coefficients α and β of the straight lines $(x, y) = (\alpha + \beta y, y)$ by means of the least squares method (see Fig. 7).

Note that after applying the least squares method, the lines describing the sowing rows may not be exactly parallel because the center of weight of a plant is not identical with the position of the seed and because seeds may slightly move during sowing from the theoretical position of the sowing row.

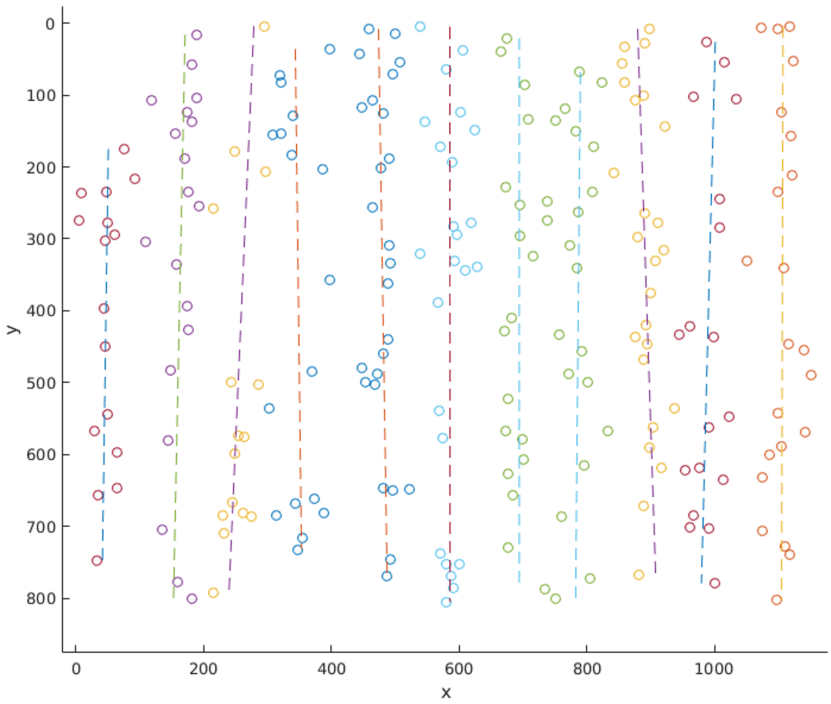

Figure 7: Sowing rows obtained with the least-squares method.

5. Test results

The method of Section 4 was tested with a set of 48 aerial photographs taken by a drone from an altitude of approximately 10 meters above a rye field, in South-Western Finland in fall 2015. The rye (cv. Evola) had been sown 25.8.2015, and the plants were 16 days old when photographed. The sowing machine (Väderstad Rapid) used a 12-cm row-to-row distance. All photography was done in the morning, between 8:50 and 9:30 o'clock. The weather was clear and the solar elevation was approximately 25°. There were no weeds on the field. The camera used was Canon EOS 5D Mark 3, with resolution 5808×3876 pixels, and the images were converted to the TIFF format before use. The direction of shooting was straight down from the drone. The vertical axis of the 2D-images was roughly parallel with the sowing rows, so that the required rotation angles were usually rather small.

In order to reduce the effect of perspective distortion, the images were partitioned into 4×4 subimages of size 1452×969 pixels, and the subimages were processed separately.

Manual evaluation of the segmentation step revealed that the excess green with Otsu thresholding performs fairly well in most pictures, although 11.5-25.2 % of the connected components contained also soil pixels in the five images analyzed manually (Table 1). In 3.3-13.9 % of the connected components, one plant was split to more than one component, and 13.8-28.5 % of the connected components contained pixels from more than one plant. 0.7 to 26.8 % of the connected components, depending on the image, contained no plant pixels at all. In overall, 68.8-92.4 % of plants in the photographs were successfully included in the connected components.

The errors made by the segmentation suggest that soil has excessively green pixels that are taken as plants by the method. Examination of the distributions of red, green and blue colors in the connected components shows that their intensity distributions may be quite similar (Images 3 and 4 in Fig. 8). However, this does not seem to correlate with errors in segmentation, as the intensity distributions suggest that Image 2 should become well segmented, but in the segmentation it has the lowest quality of the inspected five images (Table 1). Often the soil pixels that became segmented as plant material were located on the side of an earth particle and formed a longish, somewhat leaf-like form in the final segmented picture.

Table 1: Evaluation of the segmentation step. Percentages (actual numbers in parenthesis) of connected components that contained only one plant (Good components), some of them also containing soil pixels; cases where a plant is split to separate components (Split plants); connected components that contained more than one plant (Multiple plants) and connected components that did not contain plant pixels (No plants). The (%) of plant pixels in connected components lists manual estimations of 100 x (plant pixels in connected components)/(all plant pixels in the image). The last column shows the percentages of manually estimated soil pixels in connected components. The contents were calculated by manual examination of images 2, 3, 4, 14 and 35.

Image nr	Good components		Split plants	Multiple plants	No plants	(%) of plant pixels in connected components	(%) of soil pixels in connected components
	Only plant pixels	Also soil pixels					
2	43.7(80)	11.5(21)	3.3(6)	14.8(27)	26.8(49)	68.8	28.5
3	41(59)	16.7(24)	7.6(11)	20.8(30)	13.9(20)	85.5	19.3
4	30.6(45)	25.2(37)	12.9(19)	23.8(35)	7.5(11)	86.4	9.3
14	42.4(61)	14.6(21)	13.9(20)	28.5(41)	0.7(1)	88.6	4.8
35	37.3(47)	23(29)	8.7(11)	20.6(26)	10.3(13)	92.4	14.9

The errors made by the row recognizer were analysed manually (Table 2). The accuracy of the method was calculated as the ratio of correctly classified plant areas (= connected components) to the total number of plant areas. Here, the plant area was defined as the area classified as plant material by the segmentation algorithm. To aid the manual evaluation we utilized the Matlab Image Region Analyzer tool. The row recognizer was developed using Matlab programming language, and the execution times were measured by a PC (Intel Core i7 @ 8×3.6 GHz, 16 GB RAM) using Ubuntu 18.04 operating system. For practical reasons, the observed times do not include the k-means segmentation, nor the estimation of regression lines.

Fig. 9 represents a typical aerial photo from a field soon after sowing, coupled with an illustration of obtained results. The classification of individual plants into rows is indicated by using different colours.

Fig. 10 is somewhat similar, but contains rather dense groups of plants, affecting a bit the clustering of green pixels into distinct plants. A couple of

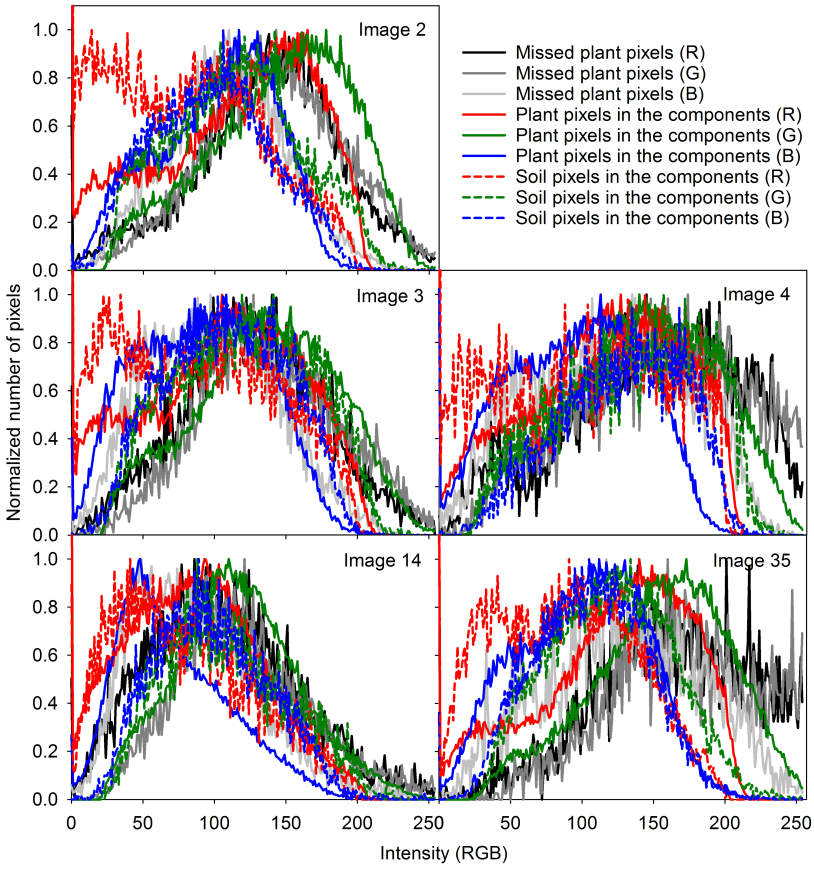

Figure 8: Distribution of red, green and blue in connected components (colored lines) and in the plant pixels not present in the components (back and gray lines) in five images. Pixels were manually classified as plant (solid lines) or soil (dashed lines).

connected components are too large, but otherwise the row detection succeeds. Some non-germinated spots can be observed, but these gaps do not essentially disturb the detection of rows.

Since the compass direction of the drone camera followed rather closely to the moving direction of the sowing machine, the rotations required to make the rows vertical were rather small. Figs. 11 and 12 were manually rotated 90 degrees in order to test the determination of the skewness of rows. The former image was an easier case, with rather smooth distribution of plants. The obtained

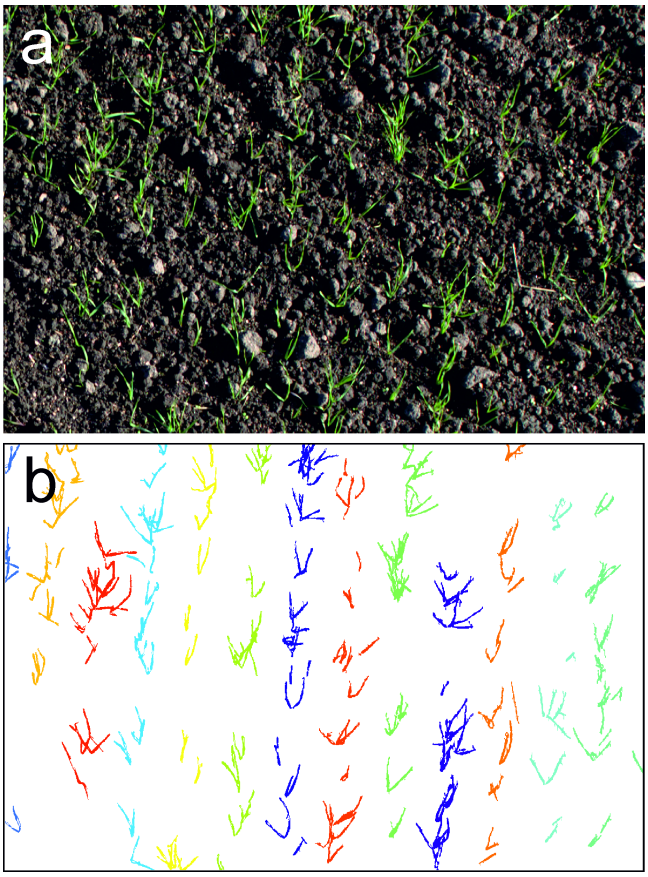

Figure 9: Test result for image 44.

classification into rows is quite close to what a human viewer would expect. The latter (Fig. 12) was a hard case, with large gaps but also quite dense spots. The result is basically acceptable, although some choices can be questioned. In fact, even for a human annotator it would be difficult to decide the correct classification, without knowing the larger context. It must be emphasized that the most of our images were easier to analyze, than these two cases.

On the basis of a qualitative inspection of row visualizations, it is justified to claim that the clustering of plants into rows in most cases matches with the decisions made by a human analyst. More detailed results concerning automatic versus manual classification are given in the appendix. The combined result of

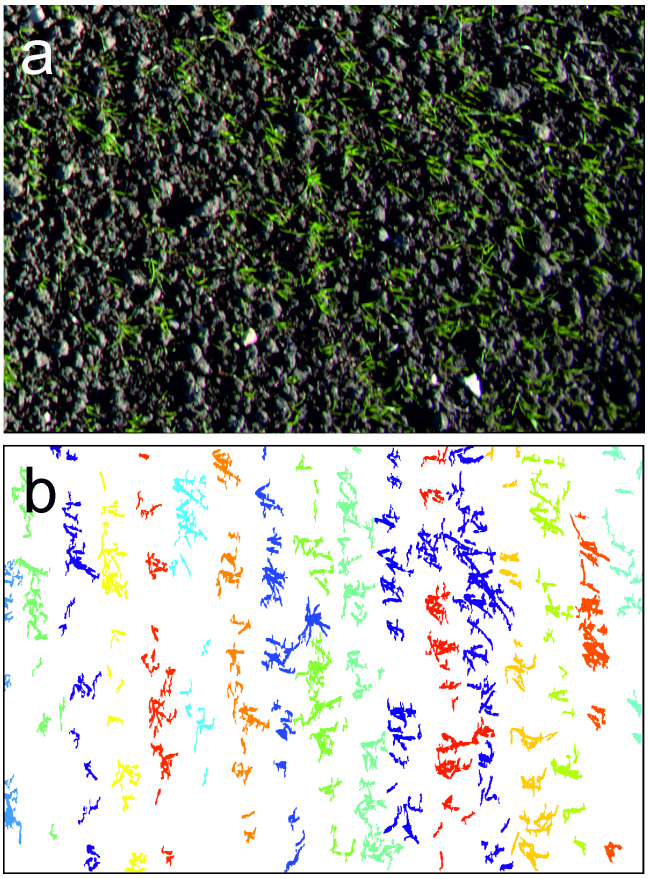

Figure 10: Test result for image 32.

manually checking all sowing row candidates in each test image was that about 94 per cent of the plants were classified into correct rows, on the average.

The processing time of a single 1452×969 image was about 0.11 seconds by our (non-optimized) implementation. After further refinement, we expect to still improve both the accuracy and the speed of the tool.

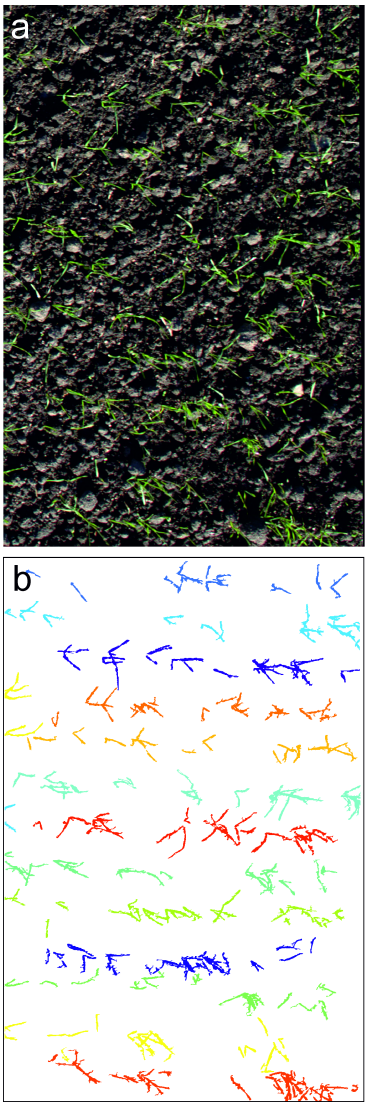

Figure 11: Test result for image 43.

Table 2: A summary of automatic versus manual analysis of sowing rows for n=48 test images. "Plants" refers to all components classified as plants by the segmentation algorithm.

	Predicted nr of rows	True nr of rows	Plants classified into wrong rows	Total no of plants	Classification accuracy(%)	Execution time (sec)
Mean	13.958	14.5	9.958	165.375	94.195	0.111
Median	14	14.5	7.5	153	94.895	0.112
SD	1.687	1.81	7.997	41.535	4.066	0.014

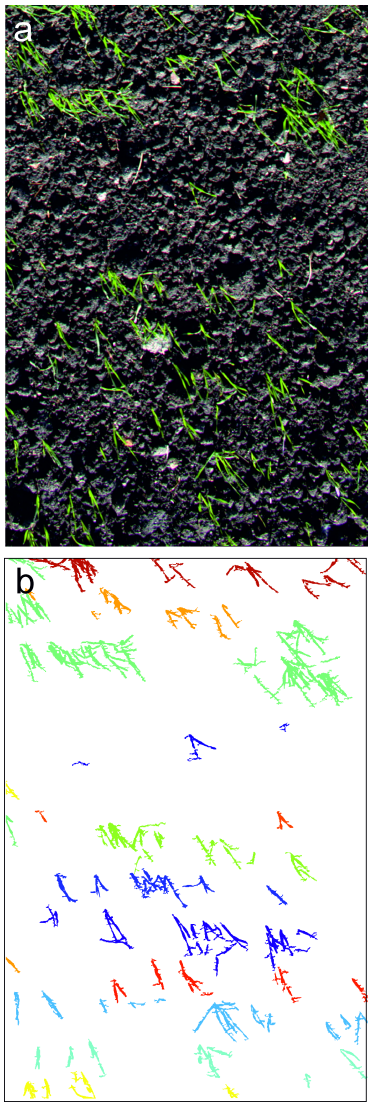

Figure 12: Test result for image 39.

6. Discussion

The results clearly show that the proposed method is capable of recognizing sparsely growing plants sown row-wise. In addition, the direction of the rows may vary to a certain extent. Also, the method makes a reliable calculation of the number of rows. Actually, the results are surprisingly good, taking into

account that the method uses basically only one geometric feature (center of gravity for the connected components) in classification. It turned out that the recognition of horizontal rows was somewhat harder than that of vertically oriented rows (due to infinite tangent of the perpendicular direction). This is easily taken care of by studying also the 90 degrees rotated photo.

The finding that the segmentation of plants with the excess green method and Otsu thresholding sometimes makes errors may suggest that the soil may contain truly excessively green pixels, at least when photographed from an UAV early in the morning when the sun is still low in the sky.

While the running time is on an acceptable level (at least for off-line processing), the current implementation leaves plenty of possibilities for speeding up, concerning both algorithms and the choice of the programming environment. There are several ways to enhance the efficiency of the recognizer. For example, the usage of a graphics processing unit would surely be of help. Another way would be rethinking of program loops by vectorization techniques.

The present study was limited to finding nearly parallel, roughly straight rows of plants, occurring in a window of restricted size. In particular, we made a restriction that the studied image is fully inside a single driving lane of the sowing machine. Handling of images covering the border area of two lanes requires a generalization. The simplest solution is to view the image as a combination of two subsets of parallel rows with the same inter-row distance, but with different offsets. In the frequency domain we would say that the dominating frequencies are divided to two phases. If the lanes are parallel, the rows can be rotated to vertical even in this case. If the lanes overlap, the border area would consist of alternating intervals of two sizes between the peaks of local maxima. If there is no overlap, a single arbitrary-size interval occurs between the lanes. We leave the detailed study of this generalization for future work.

In the present study, all plants were classified to belong to a row. The next step of the study would be to find and mark the weed candidates on the images automatically, coupled with their coordinates. The suspected weeds are plants growing in between the sowing rows, i.e. far from the line defining the row.

Assuming that sufficiently precise locations are known and marked on a map of the field, the farmer is able to remove the harmful plants.

Acknowledgement

This study was financially supported by European Agricultural Fund for Rural Development (grant nr. 30934), Academy of Finland (grant nr. 307335), University of Turku Graduate School, Turku University Foundation, and Väisälä Fund.

References

Anter, A. M., Hassenian, A. E., Oliv, D., 2019. An improved fast fuzzy c-means using crow search optimization algorithm for crop identification in agricultural. Expert Systems With Applications 118, 340354.

Bai, X., Cheng, J., Hancock, E. R., 2012. Graph-based methods in computer vision: Developments and applications. IGI Global.

Bakker, T., Wouters, H., Asselt, K., Bontsema, J., Tang, L., Müller, J., Straten, G., 2008a. A vision based row detection system for sugar beet. Computers and Electronics in Agriculture 60, 87–95.

Bakker, T., Wouters, H., Asselt, K., Bontsema, J., Tang, L., Müller, J., Straten, G., 2008b. A vision based row detection system for sugar beet. Computers and electronics in agriculture 60, 77–88.

Billingsley, J., Schoenfisch, M., 1997. The successful development of a vision guidance system for agriculture. Computers and Electronics in Agriculture 16, 147163.

Bossu, J., Gée, C., Jones, G., Truchetet, F., 2009. Wavelet transform to discriminate between crop and weed in perspective agronomic images. Computers and Electronics in Agriculture 65, 133143.

Burgos-Artizzu, X. P., Ribeiro, A., Guijarro, M., Pajares, G., 2011. Real-time image processing for crop/weed discrimination in maize fields. Computers and Electronics in Agriculture 75, 337–346.

Choi, K. H., Han, S. K., Han, S. H., Park, K., Kim, K., Kim, S., 2015. Morphology-based guidance line extraction for an autonomous weeding robot in paddy fields. Computers and Electronics in Agriculture 113, 266274.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C., 2001. Introduction to Algorithms. MIT Press and McGraw-Hill.

de Castro, A. I., Jurado-Expósito, M., Peña-Barragán, J. M., López-Granados, F., 2012. Airborne multi-spectral imagery for mapping cruciferous weeds in cereal and legume crops. Precision Agriculture 13, 302–321.

de Castro, A. I., López-Granados, F., Jurado-Expósito, M., 2013. Broad-scale cruciferous weed patch classification in winter wheat using quickbird imagery for inseason site-specific control. Precision Agriculture 14, 392–413.

García-Santillín, I., Guerrero, J. M., Montalvo, M., Pajares, G., 2018. Curved and straight crop row detection by accumulation of green pixels from images in maize fields. Precision Agriculture 19, 1841.

Gonzalez, R. C., Woods, R. E., 2002. Thresholding. In Digital Image Processing. Pearson Education.

Han, S., Zhang, Q., Ni, B., Reid, J. F., 2004. A guidance directrix approach to vision-based vehicle guidance systems. Computers and Electronics in Agriculture 43, 179195.

Huang, T., Yang, G., Tang, G., 1979. A fast two-dimensional median filtering algorithm. IEEE Trans. Acoust., Speech, Signal Processing 27, 13–18.

Jiang, G., Wang, X., Wang, Z., Liu, H., 2016. Wheat rows detection at the early growth stage based on hough transform and vanishing point. Computers and Electronics in Agriculture 123, 211223.

Jiang, G., Wang, Z., Liu, H., 2015. Automatic detection of crop rows based on multi-rois. Expert Systems with Applications 42, 2429–2441.

Josso, B., Burton, D., Lalor, M. J., 2005. Texture orientation and anisotropy calculation by fourier transform and principal component analysis. Mechanical Systems and Signal Processing 19, 1152–1161.

Kaufman, L., Rousseeuw, P. J., 1987. Clustering by means of medoids. Statistical Data Analysis Based on the L1-Norm and Related Methods, North–Holland.

Kise, M., Zhang, Q., Más, F. R., 2005. A stereovision-based crop row detection method for tractor-automated guidance. Biosystems Engineering 90, 357367.

Lloyd, S. P., 1982. Least squares quantization in pcm. IEEE Trans. on Information Theory 28, 129–137.

Maes, W. H., Steppe, K., 2018. Perspectives for remote sensing with unmanned aerial vehicles in precision agriculture. Trend Plant Sci 24, 152–163.

Marchant, J. A., Brivot, R., 1995. Real-time tracking of plant rows using a hough transform. Real-Time Imaging 1, 363371.

Motohka, T., Nasahara, K. N., ad S. Tsuchida, H. O., 2010. Applicability of greenred vegetation index for remote sensing of vegetation phenology. Remote Sensing 2, 2369–2387.

Olsen, H. L. J., 1995. Determination of row position in small-grain crops by analysis of video images. Computers and Electronics in Agriculture 12, 147162.

Otsu, N., 1975. A threshold selection method from gray-level histograms. Automatica 77, 23–27.

Pla, F., Sanchiz, J. M., Marchant, J. A., Brivot, R., 1997. Building perspective models to guide a row crop navigation vehicle. Image and Vision Computing 15, 465473.

Reid, J. F., Searcy, S. W., 1987. Automatic tractor guidance with computer vision. SAE Transactions 96, 673693.

Søgaard, H. T., Olsen, H. J., 2003. Determination of crop rows by image analysis without segmentation. Computers and Electronics in Agriculture 38, 141–158.

Southall, B., Hague, T., Marchant, J. A., Buxton, B. F., 2002. An autonomous crop treatment robot: Part i. a kalman filter model for localization and crop/weed classification. The International Journal of Robotics Research 21, 61–74.

Tang, L., Tian, L. F., 2005. Real-time crop row image reconstruction for automatic emerged corn plant spacing measurement. American Society of Agricultural and Biological Engineers 51, 1079–1087.

Vidović, I., Cupec, R., Ž. Hocenski, 2016. Crop row detection by global energy minimization. Pattern Recognition 55, 6886.

Winterhalter, W., Fleckenstein, F. V., Dornhege, C., Burgard, W., 2018. Crop row detection on tiny plants with the pattern hough transform. IEEE Robotics and Automation Letters 3, 33943401.

Woebbecke, D. M., Meyer, G. E., Bargen, K. V., Mortensen, S. A., 1995. Color indices for weed identification under various soil, residue, and lighting conditions. Transactions of the ASAE 38, 259–269.

Zhang, X., Li, X., Zhang, B., Zhou, J., Tian, G., Xiong, Y., Gu, B., 2018. Automated robust crop-row detection in maize fields based on position clustering algorithm and shortest path method. Computers and Electronics in Agriculture 154, 165175.

Appendix: Test results

Table 3: Processing time of segmentation methods over different resolution images.

Image resolution	Average execution time (sec)	
	K-means	Colour channel transform combined with Otsu's method
1452 x 969	12.225	0.041
5808 x 3876	160.931	0.867

Table 4: Automatic versus manual analysis of sowing rows.

Image nr	Predicted nr of rows	True nr of rows	Incorrectly classified plants	Total nr of plants	Classification accuracy (%)	Execution time (sec)
1	16	16	30	191	84.29	0.119
2	15	15	4	170	97.64	0.113
3	15	16	9	147	93.87	0.115
4	14	14	5	124	95.96	0.112
5	13	16	24	191	87.43	0.110
6	14	15	10	188	95.21	0.114
7	15	15	1	160	99.37	0.114
8	15	15	4	176	97.72	0.118
9	15	15	2	156	98.71	0.111
10	15	15	3	133	97.74	0.101
11	14	15	4	153	97.38	0.092
12	14	14	1	149	99.32	0.095
13	15	15	2	132	98.48	0.094
14	14	14	3	153	98.03	0.109
15	14	14	7	158	95.56	0.094
16	13	13	4	151	97.35	0.094
17	13	14	18	211	91.46	0.110
18	13	13	10	187	94.65	0.108
19	14	15	10	158	93.67	0.112
20	14	14	4	120	96.66	0.105

Table 4: Automatic versus manual analysis of sowing rows.

Image nr	Predicted nr of rows	True nr of rows	Incorrectly classified plants	Total nr of plants	Classification accuracy (%)	Execution time (sec)
21	16	16	17	267	93.63	0.141
22	15	16	6	266	97.74	0.124
23	13	15	23	147	84.35	0.102
24	16	16	12	219	94.52	0.121
25	15	17	22	194	88.65	0.124
26	18	18	3	151	98.01	0.113
27	15	18	21	228	90.78	0.126
28	15	16	22	229	90.39	0.121
29	17	18	7	175	96.00	0.115
30	16	16	10	206	95.14	0.119
31	15	17	29	227	87.22	0.122
32	17	17	4	249	98.39	0.123
33	15	14	22	202	89.1	0.137
34	13	14	24	176	86.36	0.146
35	13	14	10	126	92.06	0.117
36	13	13	7	116	93.96	0.129
37	12	13	12	120	90	0.102
38	11	12	15	151	90	0.110
39	12	12	8	96	91.66	0.117
40	13	14	6	108	94.44	0.095
41	13	13	8	137	94.16	0.122
42	12	12	7	150	95.33	0.108
43	12	13	9	139	93.52	0.119
44	13	13	2	114	98.24	0.097
45	11	12	5	131	96.18	0.087
46	11	11	0	135	100	0.088
47	13	13	8	132	93.93	0.085
48	10	10	4	139	97.12	0.088

519