Center-based clustering for line detection and application to crop rows detection

Ivan Vidović
Faculty of Electrical Engineering, University of Osijek
Cara Hadrijana 10b, HR – 31 000 Osijek, Croatia
e-mail: `ividovi2@etfos.hr`

Rudolf Scitovski[1]
Department of Mathematics, University of Osijek
Trg Lj. Gaja 6, HR – 31 000 Osijek, Croatia
e-mail: `scitowsk@mathos.hr`

Abstract. This paper proposes a new efficient method for line detection based on known incremental methods of searching for an approximate globally optimal partition of a set of data points $\mathcal{A}$ and on the `DIRECT` algorithm for global optimization. The proposed method was modified for solving the problem of detecting crop rows in agricultural production. This modification can recognize crop rows with a high accuracy, and the corresponding CPU-time is very acceptable. The method has been tested and compared on synthetic data sets with the method based on Hough transformation. The efficiency of this method might be significantly improved in direct application. The proposed method has been used in this paper for the case of two or three crop rows. The generalization to several crop rows is also given in the paper, but was not implemented. Also, the method could be expanded in case when the number of crop rows is not known in advance.

Key words: line detection; crop rows detection; clustering; incremental method; global optimization; DIRECT.

1 Introduction

Line segment or line detection problem has many applications. Let us mention one that has special importance in agricultural production. The processes of planting, fertilization, plant protection and finally harvesting are the most important processes in agriculture that can be automated. During any of the mentioned processes humans must handle a machine (a tractor, for example) with a high degree of precision and repeat the same activity for several hours, which can be very exhausting. With acceptably accurate crop rows detection it is possible to automate machine work which is usually very exhausting and sometimes too demanding for humans.

The mentioned automation processes of machines in agricultural production have been subjects of many papers. One of the first approaches to solving this problem is a Hough

[1] Corresponding author: Rudolf Scitovski, e-mail: `scitowsk@mathos.hr`, telephone number: ++385-31-224-800, fax number: ++385-31-224-801

transform-based method. (Marchant, 1996) proposed a method based on Hough transformation, which uses the information about the number of crop rows making this technique very tolerant to problems like missing plants and weeds. The method has been tested on cauliflowers, sugar beet and widely spaced double rows of wheat. (Bakker et al., 2008) transformed captured color images to gray-scale images which have good contrast between a plant and the background. Sugar beet rows have been detected using gray-scale Hough transformation. (Ji and Qi, 2011) proposed a center line crop rows detection method using gradient-based random Hough transformation. The method has been tested on sparse, general and intensive plant distribution and the results have shown that the proposed method is faster and more accurate than general Hough transform-based algorithms. An adaptation of Hough transformation applied to soil and chicory rows detection was proposed by (Leemans and Destain, 2006). They used neural networks for plants detection and adapted Hough transformation for rows detection. In this adaptation, the Hough transformation method uses theoretical crop row position and direction for reference in Hough plane. Deviation of detected crop rows from the reference was a few centimeters and authors found this compatible with the application. (Rovira-Mas et al., 2005) presented a combination of Hough transformation and connectivity analysis for finding a pathway between crop rows.

Other possibilities for crop rows detection are filter-based methods. The method for hoe guidance based on the extended Kalman filter was proposed by (Tillett and Hague, 1999). The prediction of rows position has been calculated according to the previous state and inputs using the Kalman filter and corrected by least squares incorporation of new observations. This method is very sensitive to the presence of shadows. (Olsen, 1995) proposed a method for detecting centre position of crop rows using an infrared filter based on summation of pixel gray values. The method is not sensitive to shadows while lateral winds and lateral illumination cause offset in the calculated rows position. (Hague and Tillett, 2001) proposed a combination of a (Olsen, 1995) method and a Kalman filter. The method is applicable to images with presence of shadows unlike the method presented earlier (Tillett and Hague, 1999).

There are several other approaches for crop rows detection like methods based on vanishing points or linear regression. (Pla et al., 1997) presented a method for guiding a crop rows navigation vehicle based on a scene structure building using a vanishing point. Feature extraction was done by a method based on region skeletons. The method does not work well when an image is captured at the end of the field where the remaining length of the rows is short. The method for detecting crop rows without image segmentation was proposed by (Søgaard and Olsen, 2003). Computation of line parameters was done by weighted linear regression and the method has been tested on real images. (Montalvo et al., 2012) proposed a new method for crop rows detection in maize fields with a high presence of weeds. The method is based on three steps including image segmentation, double thresholding and linear regression. The Excess Green vegetation index has been used for transforming captured RGB images to gray images and double Otsu thresholding has been applied for separating weeds and crops. For calculating line parameters associated to the crop rows, linear regression based on total least squares has been used. The main finding of this paper is double thresholding used for separating weeds and crops. The method has been favorably compared to a classical Hough approach measuring effectiveness and processing time. A new method for crop rows detection in maize fields based on linear regression and Theil-Sen estimator was proposed in (Guerrero et al., 2013). Crops and weeds are detected by using the Otsu thresholding method and the detection of crop rows is based on mapping the expected crop lines onto image and applying the Theil-Sen estimator to adjust them to the real ones.

In our paper, we first consider the problem of recognizing several lines in general position (Section 2) and propose a new cluster-based incremental method of searching for approximate lines (Subsection 2.3.2). After that, in Section 3 we propose a new method for crop rows detection as a combination of total least squares linear regression and a modification of the previously mentioned center-based incremental method. In Section 4, the proposed methods are tested and compared with the Hough transform-based algorithm on synthetic data simulating various real situations.

2 Line detection problem

Let us notice first that, without loss of generality, we can suppose that the arbitrary line in the plane is given by

$$ax + by - c = 0, \quad a^2 + b^2 = 1, \quad c \geq 0. \tag{1}$$

Let us suppose that the data point set $\mathcal{A}$ is given whose elements derive from previously unknown lines. Thereby, the number of lines can, but need not be, known in advance. On the basis of the given data point set $\mathcal{A}$, the lines could be reconstructed.

2.1 Data point set construction

The line detection problem shall first be considered on the data point set deriving from lines in general position. Let $I = \{1, \dots, m\}$ be the set of indices and

$$\mathcal{A} = \{T_i = (x_i, y_i) \in \mathbb{R}^2 \colon i \in I\} \subset R \tag{2}$$

the data point set contained in the rectangle $R = [x_{min}, x_{max}] \times [y_{min}, y_{max}]$. The data point set $\mathcal{A}$ is generated by k lines

$$p_j: \quad a_j x + b_j y - c_j = 0, \quad a_j^2 + b_j^2 = 1, \quad c_j \geq 0, \qquad j \in J = \{1, \dots, k\}, \tag{3}$$

in the following way. First, we choose interval $[y_{min}, y_{max}] \subset \mathbb{R}$ and for each $j \in J$ we define $m_j \geq 3$ equidistant spaced numbers $\eta_1, \ldots, \eta_{m_j} \in [y_{min}, y_{max}]$ and the set (see Fig.1)

$$\mathcal{A}_j = \{(\xi_i^{(j)}, \eta_i^{(j)}) + \epsilon_i^{(j)}(a_j, b_j) \colon \xi_i^{(j)} = \tfrac{1}{a_j}(c_j - b_j\eta_i^{(j)}),\ \epsilon_i^{(j)} \sim \mathcal{N}(0, \sigma^2),\ i = 1, \ldots, m_j\}.$$

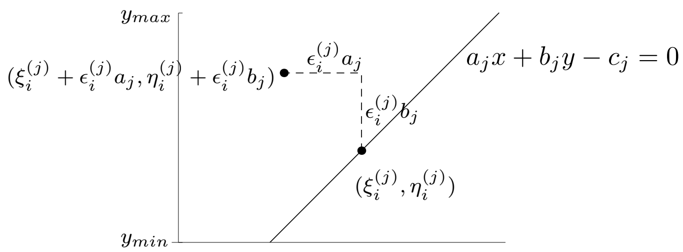

Figure 1: Data generating

The data point set $\mathcal{A} = \bigcup_{j=1}^{k} \mathcal{A}_j$ consists of $m = \sum_{j=1}^{k} m_j$ data points $T_i = (x_i, y_i) \in [x_{min}, x_{max}] \times [y_{min}, y_{max}]$, where $x_{min} = \min\limits_{i,j}\{\xi_i^{(j)} + \epsilon_i^{(j)} a_j\}$, $x_{max} = \max\limits_{i,j}\{\xi_i^{(j)} + \epsilon_i^{(j)} a_j\}$.

On the basis of the given data point set $\mathcal{A}$, lines $p_1, \ldots, p_k$ should be reconstructed.

2.2 Hough transformation method for line detection

Line detection by the Hough transform-based method (Duda and Hart, 1972; Leemans and Destain, 2006) is achieved by searching for the maximum in the Hough plane (accumulator), which represents the transformation of the input image. Each point $T = (\xi, \eta) \in \mathcal{A} \subset \mathbb{R}^2$ in the Hough plane is represented by all possible lines given in Hesse normal form passing through the point T, i.e. the set

$$\{(\alpha, \delta) \in \mathbb{R}^2 \colon \xi \cos\alpha + \eta \sin\alpha - \delta = 0\}.$$

In this way, some line p in the plane $\mathbb{R}^2$ is represented by a pair (α, δ) in the Hough plane, and some point $T \in \mathbb{R}^2$ is represented by the sequence of points in the Hough plane. Points that in the original image lie on a line increase the intensity of the point that represents that line in the Hough plane. The algorithm for line reconstruction, which is based on recognizing the most intensive points in the Hough plane will be hereinafter simply referred to as HTA.

2.3 Cluster-based line detection

Line detection problem can also be considered (Bagirov et al., 2013; Späth, 1983; Yin, 1998) as a data clustering problem of the set $\mathcal{A}$ in k nonempty disjoint subsets (clusters)

$\pi_1, \dots, \pi_k$ such that

$$\bigcup_{i=1}^{k} \pi_i = \mathcal{A}, \qquad \pi_r \cap \pi_s = \emptyset, \quad r \neq s, \qquad |\pi_j| \geq 1, \quad j = 1, \dots, k. \tag{4}$$

Such partition will be denoted by Π, and the set of all partitions of the set $\mathcal{A}$ consisting of k clusters $\pi_1, \dots, \pi_k$ will be denoted by $\mathcal{P}(\mathcal{A}; m, k)$. Clustering or grouping a data set into conceptually meaningful clusters is a well-studied problem in recent literature, and it has practical importance in a wide variety of applications such as medicine, biology, pattern recognition, facility location problem, text classification, information retrieval, earthquake investigation, understanding the Earth's climate, psychology, ranking of municipalities for financial support, business, etc. (Kogan, 2007; Liao et al., 2012; Mostafa, 2013; Pintér, 1996; Reyes et al., 2013; Sabo et al., 2011, 2013; Scitovski and Scitovski, 2013).

If we introduce the distance from the point $T = (\xi, \eta) \in \mathcal{A}$ to a line $p_j(a_j, b_j, c_j)$ given by (1) as orthogonal squared distance (Chernov, 2010; Nievergelt, 1994)

$$d(p_j(a_j, b_j, c_j), T) = (a_j \xi + b_j \eta - c_j)^2, \tag{5}$$

then to each cluster $\pi_j \in \Pi$ we can associate its center-line $\hat{p}_j(\hat{a}_j, \hat{b}_j, \hat{c}_j)$ where

$$(\hat{a}_j, \hat{b}_j, \hat{c}_j) = \operatorname*{argmin}_{a_j, b_j, c_j \in \mathbb{R}} \sum_{T \in \pi_j} d(p_j(a_j, b_j, c_j), T). \tag{6}$$

Note that, in this way, the center-line $\hat{p}_j(\hat{a}_j, \hat{b}_j, \hat{c}_j)$ of the cluster π_j is defined as the best *total least squares* (TLS) line (Chernov, 2010; Grbić et al., 2013b; Montalvo et al., 2012; Nievergelt, 1994; Scitovski et al., 1998).

After that, by introducing the objective function $\mathcal{F} \colon \mathcal{P}(\mathcal{A}; m, k) \to \mathbb{R}_+$, we can define the quality of a partition and search for the *globally optimal k-partition* by solving the following global optimization problem (GOP):

$$\operatorname*{argmin}_{\Pi \in \mathcal{P}(\mathcal{A}; m, k)} \mathcal{F}(\Pi), \qquad \mathcal{F}(\Pi) = \sum_{j=1}^{k} \sum_{T \in \pi_j} d(\hat{p}_j(\hat{a}_j, \hat{b}_j, \hat{c}_j), T), \tag{7}$$

where the center-line $\hat{p}_j$ is determined by (6).

Conversely, for a given set of center-lines $p_1, \dots, p_k$, by applying the minimal distance principle, we can define the partition $\Pi = \{\pi(p_1), \dots, \pi(p_k)\}$ of the set $\mathcal{A}$ which consists of the clusters:

$$\pi(p_j) = \{T \in \mathcal{A} : d(p_j, T) \leq d(p_s, T), \, \forall s = 1, \dots, k\}, \qquad j = 1, \dots, k, \tag{8}$$

where one has to take into account that every point of the set $\mathcal{A}$ occurs in one and only one cluster. Therefore, the problem of finding an optimal partition of the set $\mathcal{A}$ can be reduced to the following GOP

$$\operatorname*{argmin}_{a, b, c \in \mathbb{R}^k} F(a, b, c), \qquad F(a, b, c) = \sum_{T \in \mathcal{A}} \min_{1 \leq s \leq k} d(p_s(a_s, b_s, c_s), T), \tag{9}$$

where $F\colon \mathbb{R}^{3\times k} \to \mathbb{R}_+$, and $a = (a_1, \ldots, a_k)$, $b = (b_1, \ldots, b_k)$, $c = (c_1, \ldots, c_k)$. The solution of (7) and (9) coincides. Namely, it is easy to verify the following equalities

$$\begin{aligned} F(a^\star, b^\star, c^\star) &= \sum_{T\in\mathcal{A}} \min_{1\le s\le k} d(p_s^\star, T) = \sum_{j=1}^{k} \sum_{T\in\pi(p_j^\star)} \min_{1\le s\le k} d(p_s^\star, T) \\ &= \sum_{j=1}^{k} \sum_{T\in\pi(p_j^\star)} d(p_j^\star, T) = \mathcal{F}(\Pi^\star). \end{aligned} \tag{10}$$

The objective function F can have a large number of independent variables. It does not have to be either convex or differentiable, but it is a Lipschitz continuous function (Grbić et al., 2013a; Pintér, 1996; Sabo et al., 2013). The objective function F can also be considered as a symmetric function $\Phi\colon \mathbb{R}^{k\times 3} \to \mathbb{R}_+$, $\Phi(\zeta_1, \ldots, \zeta_k) := F(a, b, c)$, where $\zeta_j = (a_j, b_j, c_j)$. Because of the symmetry property, the function Φ has at least $k!$ local and global minimizers. Therefore, this becomes a complex GOP for a symmetric Lipschitz continuous function.

The following example gives a solution to the problem (9) in the simplest case $k = 1$.

Example 1. *For the given data point set* $\mathcal{A} = \{T_i = (x_i, y_i) \in \mathbb{R}^2 \colon i \in I\} \subset R$, $R = [x_{min}, x_{max}] \times [y_{min}, y_{max}]$, *by solving GOP* (6) *the corresponding TLS line should be determined. The best TLS line passes through the centroid of the data* (x_c, y_c), $x_c = \frac{1}{m}\sum_{i=1}^{m} x_i$, $y_c = \frac{1}{m}\sum_{i=1}^{m} y_i$ *and it can be determined (Chernov, 2010; Nievergelt, 1994) by the eigenvector which belongs to a smaller eigenvalue of the matrix* $B^T B$, *where*

$$B = \begin{bmatrix} x_1 - x_c & y_1 - y_c \\ \vdots & \vdots \\ x_m - x_c & y_m - y_c \end{bmatrix}.$$

We will mention several known methods of searching for a globally optimal partition or an approximate globally optimal partition, which can be adapted for solving GOP (9). Since our objective function F in (9) is a Lipschitz continuous function, there are numerous methods for its minimization (Evtushenko, 1985; Floudas and Gounaris, 2009; Neumaier, 2004; Pintér, 1996). One of the most popular algorithms for solving a GOP for a Lipschitz continuous function is the `DIRECT` (`DI`viding `RECT`angles) algorithm (Finkel, 2003; Gablonsky, 2001; Jones et al., 1993). However, a large number of independent variables of the minimizing function F and a large number of its global minimizers make these methods insufficiently efficient.

Instead of searching for the globally optimal partition, various simplifications are often proposed in the literature that would find a good partition. However, we usually do not know how close this partition is to the globally optimal one.

2.3.1 Adjustment of the k-means algorithm

The most popular algorithm of searching for a locally optimal partition is a well-known k-means algorithm (see e.g. Kogan (2007); Liao et al. (2012); Scitovski and Sabo (2014); Späth (1983); Teboulle (2007)). If we have a good initial approximation, this algorithm can provide an acceptable solution. In case we do not have a good initial approximation, the algorithm should be restarted with various random initializations, as proposed by (Leisch, 2006). This algorithm was modified by Späth (1981) for solving GOP (9) (see Algorithm 1).

Algorithm 1 (k-means algorithm)

1. Let $\mathcal{A} = \{T_i = (x_i, y_i) : i = 1, \dots, m\}$, and $p_1, \dots, p_k$ be mutually different lines;
2. By using the minimal distance principle (8) determine k disjoint unempty clusters $\pi_1(p_1), \dots, \pi_k(p_k)$;
3. For each cluster π_j define center-line $\hat{p}_j$ according to (6);
4. **if** $\{\hat{p}_1, \dots, \hat{p}_k\} = \{p_1, \dots, p_k\}$ **then**
5.   STOP
6. **else**
7.   set $p_j := \hat{p}_j$, $j = 1, \dots, k$ and go to Step 2;

2.3.2 Adjustment of incremental methods

The next possibility of searching for the solution of (9) is adjustment of incremental methods of searching for an approximate globally optimal partition (Likas et al., 2003; Bagirov and Ugon, 2005; Scitovski and Scitovski, 2013). This adjustment will be called *Incremental Method for Line Detection* (IMLD).

First, suppose that the initial center-line $\hat{p}_1(\hat{a}_1, \hat{b}_1, \hat{c}_1)$ is chosen or that we have taken it to be the best TLS line (see Example 1). The next center-line $\hat{p}_2(\hat{a}_2, \hat{b}_2, \hat{c}_2)$ will be determined by solving the following GOP (see Scitovski and Scitovski (2013)):

$$\underset{\alpha,\beta,\gamma\in\mathbb{R}}{\operatorname{argmin}} \sum_{i=1}^{m} \min\{d(\hat{p}_1(\hat{a}_1, \hat{b}_1, \hat{c}_1), T_i), d(p(\alpha,\beta,\gamma), T_i)\}, \quad \alpha^2+\beta^2=1,\ \gamma \geq 0,$$

by using the `DIRECT` algorithm. After that, an approximate globally optimal partition $\Pi^\star = \{\pi_1^\star(p_1^\star), \pi_2^\star(p_2^\star)\}$ with center-lines $p_1^\star, p_2^\star$ will be obtained by using the k-means algorithm (Algorithm 1) where we take $(\hat{p}_1, \hat{p}_2)$ as initial center-lines.

Generally, if the first $k-1$ center-lines $\hat{p}_1, \dots, \hat{p}_{k-1}$ are known, the center-line $\hat{p}_k(\hat{a}_k, \hat{b}_k, \hat{c}_k)$ will be determined by solving the following GOP

$$\underset{\alpha,\beta,\gamma\in\mathbb{R}}{\operatorname{argmin}} \sum_{i=1}^{m} \min\{\hat{\delta}_{k-1}^i, d(p(\alpha,\beta,\gamma), T_i)\}, \quad \hat{\delta}_{k-1}^i = \min\{d(\hat{p}_1, T_i), \dots, d(\hat{p}_{k-1}, T_i)\}, \tag{11}$$

by using the `DIRECT` algorithm. After that, an approximate globally optimal partition $\Pi^\star = \{\pi_1^\star(p_1^\star), \ldots, \pi_k^\star(p_k^\star)\}$ with center-lines $p_1^\star, \ldots, p_k^\star$ will be obtained by using Algorithm 1 where we take $(\hat{p}_1, \ldots, \hat{p}_k)$ as initial center-lines.

The following example illustrates HTA and the proposed IMLD for line detection.

Example 2. *On the basis of the given lines:* $p_1 : 0.9x - 0.4y + 0.18 = 0$, $p_2 : x + 0.05y + 0.6 = 0$, $p_3 : 0.135x + y + 0.3 = 0$, *data point set* $\mathcal{A}$ *is constructed as in Section 2.1 (see Fig. 2a). It can be noticed that IMLD has recognized all three lines (see Fig. 2c), whereas HTA did not recognize line* p_3 *(see Fig. 2b).*

(a) Generated data points (b) HTA (c) IMLD

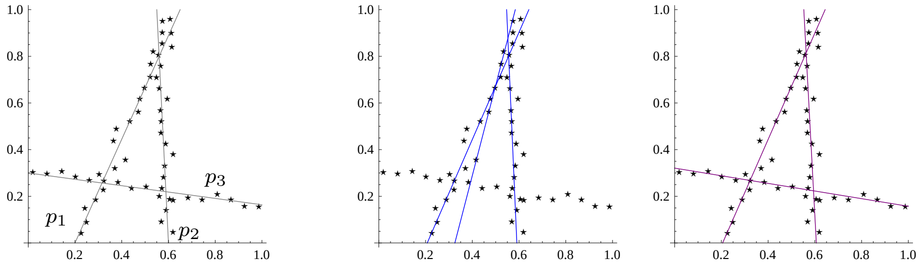

Figure 2: Line detection by HTA and IMLD

Remark 1. Note that the proposed IMLD method in each iteration searches for the next center-line as a globally optimal solution of the problem (11) by using `DIRECT` algorithm for global optimization. A similar method was proposed by Bagirov et al. (2013), but in each step of iterative process, the next center-line is searched for by solving corresponding locally optimal problem. Thereby, special attention was paid to the choice of good initial approximation. Since, in this case, the objective function may have several local and global minima, one cannot know in advance how near this solution is to the globally optimal one.

3 An application: crop rows detection

A special case of the line detection problem is a crop rows detecting problem, which has already been mentioned in the Introduction. Let us first formally define the problem. Suppose that the data point set $\mathcal{A} \subset R$, $R = [x_{min}, x_{max}] \times [y_{min}, y_{max}]$ is generated as in Subsection 2.1 on the basis of lines $p_1, \ldots, p_k$ that intersect the interval $[x_{min}, x_{max}]$ in equidistant knots $\nu_1, \ldots, \nu_k \in [x_{min}, x_{max}]$ and have the common vanishing point $B = (x_B, y_B)$, $x_{min} \le x_B \le x_{max}$, $y_B > y_{max}$ (Leemans and Destain, 2006) (see e.g. Figures 4a,5a,6a,7a). On the basis of the data point set $\mathcal{A}$, lines $p_1, \ldots, p_k$ should be reconstructed.

As stated in the Introduction, there are many different approaches to solving this problem that can be found in literature. The problem can also be solved by using the IMLD method mentioned in Section 2.3.2. Numerical experiments carried out in Section 4 have shown that IMLD does not always offer the best results, and that additional modifications need to be done. These modifications were described in algorithms mentioned below. The algorithms for solving the problem of detecting two and three crop rows will be mentioned specially. These procedures can easily be generalized for several crop rows.

3.1 The algorithm for solving two crop rows detecting problem

On the basis of the given data point set $\mathcal{A}$, the following algorithm gives a satisfactory approximation of two globally optimal crop rows. The algorithm includes searching for a TLS line (three times) and the k-means algorithm (Algorithm 1).

Algorithm 2 (approximate globally optimal 2-partition)

1. Let $\mathcal{A} = \{T_i = (x_i, y_i) : i = 1, \ldots, m\}$;
2. For the data point set $\mathcal{A}$ determine the best TLS line p_0;
3. By using line p_0 divide the set $\mathcal{A}$ into two disjoint subsets such that $\mathcal{A} = \mathcal{A}_1 \cup \mathcal{A}_2$;
4. **for** $j = 1, 2$ **do**
5. For the data point set $\mathcal{A}_j$ determine the best TLS line $\hat{p}_j$;
6. **end for**
7. Apply Algorithm 1 to data set $\mathcal{A}$ with initial center-lines $\hat{p}_1, \hat{p}_2$

3.2 The algorithm for solving three crop rows detecting problem

On the basis of the given data point set $\mathcal{A}$, the following algorithm gives a satisfactory approximation of three globally optimal crop rows. The algorithm includes searching for a TLS line, solving a GOP by using the `DIRECT` algorithm and Algorithm 1.

Algorithm 3 (approximate globally optimal 3-partition)

1. Let $\mathcal{A} = \{T_i = (x_i, y_i) : i = 1, \ldots, m\}$;
2. For the data point set $\mathcal{A}$ determine the best TLS line $\hat{p}_0$;
3. By using line $\hat{p}_0$ divide the set $\mathcal{A}$ into two disjoint subsets such that $\mathcal{A} = \mathcal{A}_1 \cup \mathcal{A}_2$;
4. **for** $j = 1, 2$ **do**
5. By using the `DIRECT` method solve the following GOP
$$(\zeta_1, \zeta_2, \zeta_3) = \operatorname*{argmin}_{\alpha,\beta,\gamma \in \mathbb{R}} \sum_{T \in \mathcal{A}_j} \min\{d(\hat{p}_0, T), d(p(\alpha, \beta, \gamma), T)\};$$
6. and set $\hat{p}_j := p(\zeta_1, \zeta_2, \zeta_3)$
7. **end for**
8. Apply Algorithm 1 to data set $\mathcal{A}$ with initial center-lines $\hat{p}_0, \hat{p}_1, \hat{p}_2$

3.3 The algorithm for solving several crop rows detecting problem

Based on given data set $\mathcal{A}$, a satisfactory approximation of $k \geq 2$ globally optimal crop rows should be determined. If $k = 2$, Algorithm 2 can be used, and if $k = 3$, Algorithm 3 can be used. Generally, for $k \geq 2$, a satisfactory approximation of k globally optimal crop rows can be obtained by using Algorithm 4. In the construction of Algorithm 4, Algorithm 2 and Algorithm 3 were used several times.

Algorithm 4 (approximate globally optimal k-partition)

1. Let $\mathcal{A}_0 = \{T_i = (x_i, y_i) \colon i = 1, \ldots, m\}$; $k \geq 2$; $n = k$; $cnt = 0$; $\Pi = \emptyset$; $\hat{\Pi} = \emptyset$;
2. **while** $n > 3$ **do**
3. **for** $j = 0$ **to** $j < 2^{cnt}$ **do**
4. For data set $\mathcal{A}_j$ determine the best TLS line p_j; Add p_j to Π;
5. By using line p_j divide the set $\mathcal{A}_j$ into two disjoint subsets $\mathcal{B}_{2j}, \mathcal{B}_{2j+1}$;
6. **end for**
7. **if** $n \bmod 2 \neq 0$ **then**
8. Add Π to $\hat{\Pi}$
9. **end if**
10. Set $\Pi = \emptyset$; $n = \lfloor n/2 \rfloor$; $cnt = cnt + 1$;
11. **for** $j = 0$ **to** $j < 2^{cnt}$ **do**
12. $\mathcal{A}_j = \mathcal{B}_j$; $\mathcal{B}_j = \emptyset$;
13. **end for**
14. **end while**
15. **for** $j = 0$ **to** $j < 2^{cnt}$ **do**
16. **if** $n == 2$ **then**
17. Apply steps 2-6 of Algorithm 2 to data set $\mathcal{A}_j$; Add obtained p_1, p_2 to $\hat{\Pi}$;
18. **end if**
19. **if** $n == 3$ **then**
20. Apply steps 2-7 of Algorithm 3 to data set $\mathcal{A}_j$; Add obtained p_0, p_1, p_2 to $\hat{\Pi}$;
21. **end if**
22. **end for**
23. Apply Algorithm 1 to data set $\mathcal{A}_0$ with the set of initial center-lines $\hat{\Pi}$;

4 Numerical experiments

The proposed algorithms for crop rows detection will be compared to each other and also to the HTA algorithm.

Let $P = \{p_1, \ldots, p_k\}$ be the set of original lines and let $P^\star = \{p_1^\star, \ldots, p_k^\star\}$ be the set of reconstructed lines. The quality of reconstruction will be measured by using Hausdorff distance between the sets P and $P^\star$

$$\hat{d}_H := d_H(P, P^\star) = \max\{\max_r \min_s d_l(p_r, p_s^\star), \max_s \min_r d_l(p_r, p_s^\star)\}, \quad r, s \in \{1, \ldots, k\}. \tag{12}$$

Thereby, the distance d_l between two lines

$$p_1 : \; a_1 x + b_1 y - c_1 = 0, \quad p_2 : \; a_2 x + b_2 y - c_2 = 0,$$

can be defined in the following ways.

(i) *Angle distance* $d_A(p_1, p_2)$ is according to Cupec et al. (2009) defined by

$$d_A(p_1, p_2) = \begin{cases} 1 - |\vec{n}_1 \cdot \vec{n}_2|, & \text{if } |\vec{n}_1 \cdot \vec{n}_2| < 1, \\ \frac{|c_1 - (\vec{n}_1 \cdot \vec{n}_2) c_2|}{1 + |c_1 - (\vec{n}_1 \cdot \vec{n}_2) c_2|}, & \text{if } |\vec{n}_1 \cdot \vec{n}_2| = 1, \end{cases} \qquad \begin{array}{l} \vec{n}_1 = a_1 \vec{i} + b_1 \vec{j}, \\ \vec{n}_2 = a_2 \vec{i} + b_2 \vec{j}. \end{array} \tag{13}$$

(ii) *Integral distance* $d_I(p_1, p_2)$ represents the area between the lines p_1, p_2 in the data area $[x_{min}, x_{max}] \times [y_{min}, y_{max}]$

$$d_I(p_1, p_2) = \int_{y_{min}}^{y_{max}} |x_2(y) - x_1(y)| dy, \tag{14}$$

where $x_1(y) = \frac{1}{a_1}(c_1 - b_1 y)$, $x_2(y) = \frac{1}{a_2}(c_2 - b_2 y)$. It can be shown that there holds

$$d_I(p_1, p_2) = (y_{max} - y_{min}) \left| \alpha - \frac{\beta^2}{2}(y_{max} + y_{min}) \right|, \quad \alpha = \frac{c_1}{a_1} - \frac{c_2}{a_2}, \quad \beta = \frac{b_1}{a_1} - \frac{b_2}{a_2}. \tag{15}$$

Example 3. *For pairs of lines in Fig. 3, both their angle (d_A) and integral (d_I) distances are shown in Table 1.*

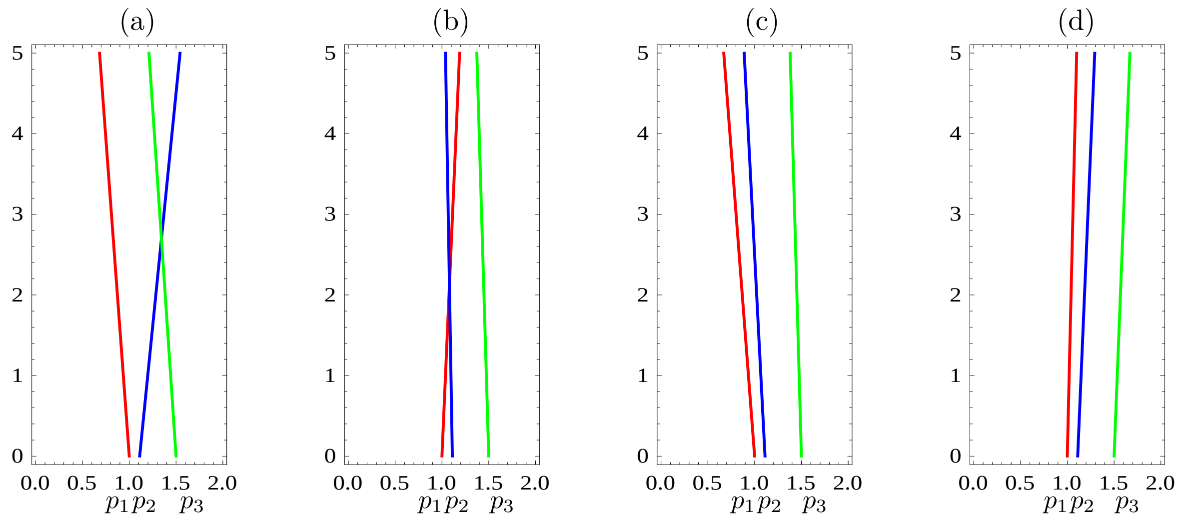

Figure 3: Angle and integral distances between lines

	(Fig. 3a)		(Fig. 3b)		(Fig. 3c)		(Fig. 3d)	
	d_A	d_I	d_A	d_I	d_A	d_I	d_A	d_I
$d_l(p_1, p_2)$	.01116	2.43	.00137	0.33	.0002	0.82	.00013	0.76
$d_l(p_1, p_3)$	.00001	2.57	.00199	1.71	.0009	3.02	.00009	2.67
$d_l(p_2, p_3)$	.01035	0.91	.00006	1.81	.0002	2.20	0	1.91

Table 1: Comparison of angle and integral distances between lines

As can be seen from the above example, unlike the integral distance, the angle distance does not recognize the distance between almost parallel lines well enough. For that reason we will use the integral distance (15) in Hausdorff distance (12) for the distance between two lines $d_l(p_r, p_s^\star)$.

The proposed algorithms for crop rows detection, Algorithm 2, Algorithm 3 and IMLD will be compared with HTA on synthetic data, which are constructed at the beginning of Section 3. Let us first choose the vanishing point $B \in [0,1] \times [2,10]$ and k lines which pass through the vanishing point B and intersect the interval $[0,1]$ in k equidistantly distributed points $0 < \nu_1 < \nu_2 < \cdots < \nu_k < 1$, where $\nu_{j+1} - \nu_j = \delta < \frac{1}{k-1}$. For example, the point ν_1 can be chosen in subinterval $[0, 1-(k-1)\delta]$, and other points are then $\nu_j = \nu_1 + (j-1)\delta$, $j = 2, \ldots, k$. After that, similarly to Subsection 2.1, using the line $p_j : a_j x + b_j y - c_j = 0$, $a_j^2 + b_j^2 = 1$ passing through the point $\{B, \nu_j\}$, the set

$$\mathcal{A}_j = \{(\xi_i^{(j)}, \eta_i^{(j)}) + \epsilon_i^{(j)}(a_j, b_j) \colon \xi_i^{(j)} = \tfrac{1}{a_j}(c_j - b_j \eta_i^{(j)}),\ \epsilon_i^{(j)} \sim \mathcal{N}(0, \sigma^2),\ i = 1, \ldots, m_j\}, \quad (16)$$

will be determined, where $\eta_1, \ldots, \eta_{m_j} \in [0,2]$ are equidistantly spaced numbers. Data point set $\mathcal{A}$ is then

$$\mathcal{A} = \bigcup_{j=1}^{k} \left(\mathcal{A}_j \cap [0,1] \times [0,2]\right), \quad |\mathcal{A}| = m,$$

and their elements will be denoted by $T_i = (x_i, y_i)$, $i = 1, \ldots, m$.

The comparison will be carried out for "complete sowing" and for "incomplete sowing" in case of relatively small variance $\sigma^2 = 0.002$ and also in case of 10-times greater variance $\sigma^2 = 0.02$. By "complete sowing" we mean sowing where all the plants have sprung up, and by "incomplete sowing" we mean sowing where some of the plants have not sprung up. In our numerical experiments, complete sowing has been simulated in the way that sets $\mathcal{A}_j$ given by (16) contain 20 points, and incomplete sowing means that $5-25\%$ randomly chosen points have been dropped from these sets.

In each numerical experiment the Hausdorff distance (12) between the set of original lines and the set of detected lines obtained by applying HTA, IMLD, and Algorithm 2 (i.e. Algorithm 3) will be determined. Thereby, the distance between the lines p_r and $p_s^\star$ in (12) as an integral distance (14) will be determined.

The efficiency of considered algorithms will be measured by the used CPU-time.

4.1 Comparison for $k = 2$ crop rows

The experiment of choosing and reconstructing two crop rows, as described previously, will be carried out 100 times with $m_1 = m_2 = 20$, whereby the vanishing point B has been randomly chosen in $[0, 1] \times [2, 10]$.

Algorithm	Complete sowing HTA	IMLD	Algorithm 2	Incomplete sowing HTA	IMLD	Algorithm 2
$\hat{d}_H < 0.005$	60	75	100	54	81	100
$0.005 \leq \hat{d}_H < 0.01$	39	-	-	46	-	-
$0.01 \leq \hat{d}_H < 0.02$	1	-	-	-	-	-
$0.02 \leq \hat{d}_H < 0.20$	-	25	-	-	19	-
CPU-time (sec)	1.25	.23	.04	1.25	.23	.04

Table 2: Testing the methods for solving the problem of detecting two crops with $\sigma^2 = 0.002$

Table 2 shows the results of testing the observed methods for solving the problem of detecting two crop rows on data generated in the aforementioned way in case of relatively small variance $\sigma^2 = 0.002$. As can be seen in Table 2, the Algorithm 2 always recognizes crop rows with a very high accuracy. IMLD attains the same recognition accuracy in about 80% cases, while in the rest 20% of cases the recognition accuracy is worse. The HTA attains the same high recognition accuracy in somewhat more than 50% cases, while in the rest of the cases the recognition accuracy is worse.

Furthermore, as can be seen from Table 2, the average used CPU-time for Algorithm 2 per experiment is 0.04 sec and it is 5 times shorter than the used CPU-time for the IMLD and 30 times shorter than the used CPU-time for HTA.

It is interesting to notice that in case of two crop rows detection there are no significant differences in the application in all of the observed methods with complete and incomplete sowing.

Algorithm	Complete sowing HTA	IMLD	Algorithm 2	Incomplete sowing HTA	IMLD	Algorithm 2
$\hat{d}_H < 0.05$	31	84	99	30	69	98
$0.05 < \hat{d}_H \leq 0.1$	12	3	1	14	4	2
$0.1 < \hat{d}_H \leq 0.2$	9	12	-	17	27	-
$0.2 < \hat{d}_H \leq 0.5$	48	1	-	39	-	-
CPU-time (sec)	1.25	.24	.04	1.24	.23	.04

Table 3: Testing of the methods for solving the problem of detecting two crop rows with $\sigma^2 = 0.02$

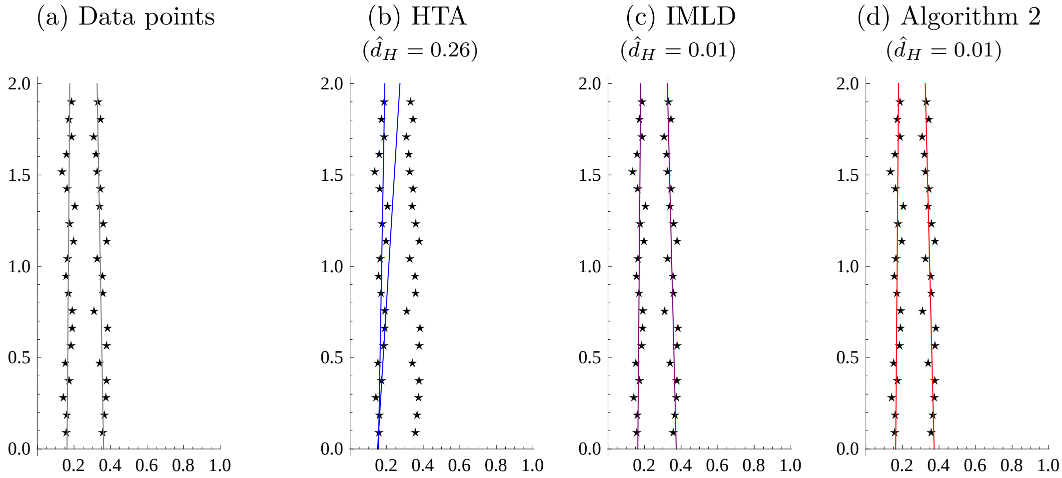

Figure 4: An example of a numerical test of methods for solving the problem of detecting two crop rows for data with variance $\sigma^2 = 0.02$ that simulate complete sowing

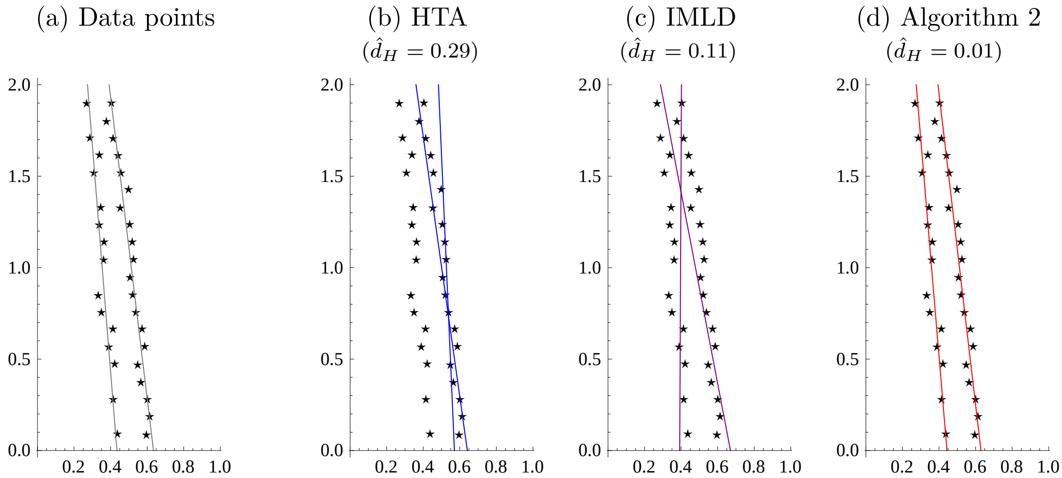

Figure 5: An example of a numerical test of methods for solving the problem of detecting two crop rows for data with variance $\sigma^2 = 0.02$ that simulate incomplete sowing

Also, on the basis of data generated in the previously described way, in case of 10-times greater variance $\sigma^2 = 0.02$, Algorithm 2 has attained good performance: high accuracy (see Table 3 and Fig. 4d and Fig. 5d) and very short used CPU-time (see Table 3).

IMLD attains the same recognition accuracy in 70% to 85% cases (as in Fig. 4c), while in other cases the recognition accuracy is worse (see Table 3). One such experiment with complete sowing and good recognition is shown in Fig. 4c, and one experiment with incomplete sowing and bad recognition is shown in Fig. 5c.

HTA attains the same high recognition accuracy only in 30% cases, while in other cases the recognition accuracy is worse (see Table 3). One such experiment with complete sowing and bad recognition is shown in Fig. 4b, and one experiment with incomplete sowing and bad recognition is shown in Fig. 5b.

Furthermore, as can be seen from Table 3, in case of relatively high variance $\sigma^2 = 0.02$, the used CPU-time for all methods has not changed.

4.2 Comparison for $k = 3$ crop rows

The experiment of choosing and reconstructing three crop rows in the previously described way will be carried out 100 times with $m_1 = m_2 = m_3 = 20$, whereby the vanishing point B is randomly chosen in $[0, 1] \times [2, 10]$.

Table 4 shows the results of testing the observed methods for solving the problem of detecting three crop rows on the basis of data generated in the previously mentioned way in case of relatively small variance $\sigma^2 = 0.002$. It can be seen that Algorithm 3 as well as HTA always recognizes crop rows with a very high accuracy. IMLD attains the same recognition accuracy in 80% cases by complete sowing and in 60% cases by incomplete sowing, while in other cases the recognition accuracy is worse.

	Complete sowing			Incomplete sowing		
Algorithm	HTA	IMLD	Algorithm 3	HTA	IMLD	Algorithm 3
$\hat{d}_H < 0.05$	100	79	100	100	63	100
$0.05 < \hat{d}_H \leq 0.1$	-	6	-	-	5	-
$0.1 < \hat{d}_H \leq 0.15$	-	13	-	-	16	-
$0.15 < \hat{d}_H \leq 0.20$	-	2	-	-	16	-
CPU-time (sec)	1.25	.45	.37	1.25	.44	.36

Table 4: Testing the methods for solving the problem of detecting three crop rows with $\sigma^2 = 0.002$

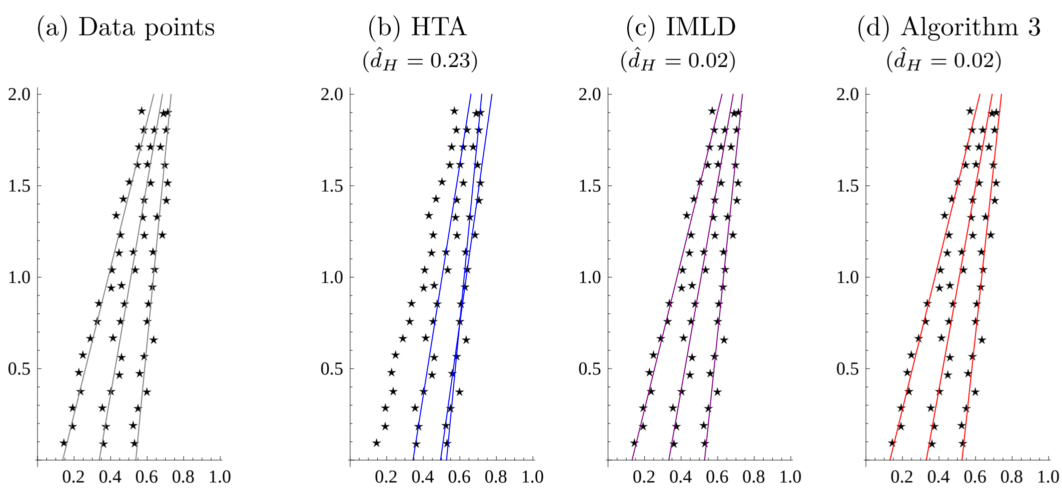

Figure 6: An example of a numerical test of methods for solving the three crop rows detection problem for data with variance $\sigma^2 = 0.02$ that simulate complete sowing

Furthermore, as can be seen from Table 4, the CPU-time for IMLD and Algorithm 3 is very acceptable, while the used CPU-time for HTA is $3-4$ times longer.

Again, in the case of recognizing three crop rows, there are no significant differences in the application of all observed methods with complete and incomplete sowing.

Algorithm	Complete sowing HTA	Complete sowing IMLD	Complete sowing Algorithm 3	Incomplete sowing HTA	Incomplete sowing IMLD	Incomplete sowing Algorithm 3
$\hat{d}_H < 0.05$	18	74	100	8	67	98
$0.05 < \hat{d}_H \leq 0.1$	5	2	-	5	3	2
$0.1 < \hat{d}_H \leq 0.15$	2	10	-	6	11	-
$0.15 < \hat{d}_H \leq 0.20$	6	14	-	4	19	-
CPU-time (sec)	1.25	.46	.36	1.25	.45	.36

Table 5: Testing the methods for solving the problem of detecting three crop rows with $\sigma^2 = 0.02$

Also, on the basis of data generated in the previously mentioned way, in case of 10-times greater variance $\sigma^2 = 0.02$, Algorithm 3 has attained good performances: the high recognition accuracy (see Table 5 and Fig. 6d and Fig. 7d) and very short used CPU-time (see Table 5).

IMLD attains the same recognition accuracy in about 70% cases (as in Fig. 6c), while in other cases the recognition accuracy is worse (see Table 5). One such experiment with complete sowing and good recognition is shown in Fig. 6c, and one experiment with incomplete sowing and bad recognition is shown in Fig. 7c.

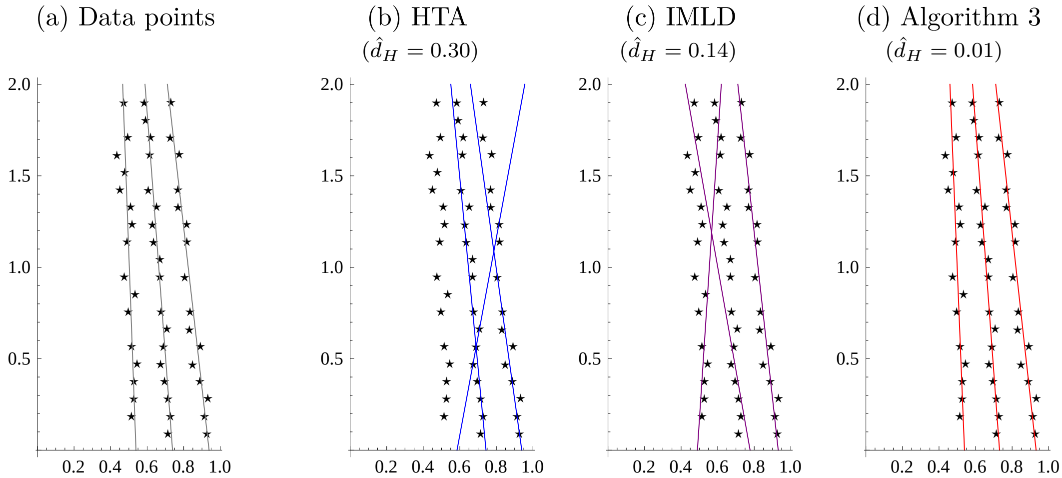

Figure 7: An example of a numerical test of methods for solving the problem of detecting three crop rows for data with variance $\sigma^2 = 0.02$ that simulate incomplete sowing

HTA attains the same recognition accuracy only in a small number of cases, while in most experiments the recognition accuracy is worse (see Table 5). Also, a large number of bad cases has not even been registered in Table 5. One such experiment with complete sowing and bad recognition is shown in Fig. 6b, and one experiment with incomplete sowing and bad recognition is shown in Fig. 7b.

Furthermore, in case of relatively high variance $\sigma^2 = 0.02$, the used CPU-time for all methods has not changed, as can be seen from Table 5.

5 Conclusion

The proposed method for crop rows detection described in Algorithm 2 and Algorithm 3 recognizes crop rows with a very high accuracy. Thereby, the used CPU-time is very acceptable from the application point of view. For this reason, we suppose that the proposed method would be very acceptable in a direct application. However, in that case the used CPU-time could be even shorter such that Algorithm 2, i.e. Algorithm 3, is performed only at the beginning of the working process. After that, it is sufficient to perform corrections by k-means Algorithm 1 successively.

The generalization to several crop rows is also given in the paper, but was not implemented. Also, this method could be extended for the case when the number of lines is not known in advance. In that case, some of the known indexes should be adjusted (see e.g. Vendramin et al. (2009)).

Acknowledgements.

The authors would like to thank to anonymous referees and journal editors for their careful reading of the paper and insightful comments that helped us improve the paper. This work is supported by the Ministry of Science, Education and Sports, Republic of Croatia, through research grants `235-2352818-1034` and `165-0361621-2000`.

References

Bagirov, A.M., Ugon, J., 2005. An algorithm for minimizing clustering functions. Optimization 54, 351–368.

Bagirov, A.M., Ugon, J., Mirzayeva, H., 2013. Nonsmooth nonconvex optimization approach to clusterwise linear regression problems. European Journal of Operational Research 229, 132–142.

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

Chernov, N., 2010. Circular and linear regression: Fitting circles and lines by least squares. volume 117 of *Monographs on Statistics and Applied Probability*. Chapman & Hall/CRC.

Cupec, R., Grbić, R., Sabo, K., Scitovski, R., 2009. Three points method for searching the best least absolute deviations plane. Applied Mathematics and Computation 215, 983–994.

Duda, R.O., Hart, P.E., 1972. Use of the Hough Transformation to detect lines and curves in pictures. Communications of the ACM 15, 11–15.

Evtushenko, Y.G., 1985. Numerical Optimization Techniques (Translations Series in Mathematics and Engineering). Springer-Verlag, Berlin.

Finkel, D.E., 2003. DIRECT Optimization Algorithm User Guide. Center for Research in Scientific Computation. North Carolina State University. http://www4.ncsu.edu/definkel/research/index.html.

Floudas, C.A., Gounaris, C.E., 2009. A review of recent advances in global optimization. Journal of Global Optimization 45, 3–38.

Gablonsky, J.M., 2001. DIRECT Version 2.0. Technical Report. Center for Research in Scientific Computation. North Carolina State University.

Grbić, R., Nyarko, E.K., Scitovski, R., 2013a. A modification of the DIRECT method for Lipschitz global optimization for a symmetric function. Journal of Global Optimization 57, 1193–1212.

Grbić, R., Scitovski, K., Sabo, K., Scitovski, R., 2013b. Approximating surfaces by the moving least absolute deviations method. Applied Mathematics and Computation 219, 4387–4399.

Guerrero, J., Guijarro, M., Montalvo, M., Romeo, J., Emmi, L., Ribeiro, A., Pajares, G., 2013. Automatic expert system based on images for accuracy crop row detection in maize fields. Expert Systems with Applications 40, 656–664.

Hague, T., Tillett, N., 2001. A bandpass filter-based approach to crop row location and tracking. Mechatronics 11, 1–12.

Ji, R., Qi, L., 2011. Crop-row detection algorithm based on Random Hough Transformation. Mathematical and Computer Modelling 54, 1016–1020.

Jones, D.R., Perttunen, C.D., Stuckman, B.E., 1993. Lipschitzian optimization without the Lipschitz constant. Journal of Optimization Theory and Applications 79, 157–181.

Kogan, J., 2007. Introduction to Clustering Large and High-dimensional Data. Cambridge University Press.

Leemans, V., Destain, M.F., 2006. Line cluster detection using a variant of the Hough transform for culture row localisation. Image and Vision Computing 24, 541–550.

Leisch, F., 2006. A toolbox for k-centroids cluster analysis. Computational Statistics & Data Analysis 51, 526–544.

Liao, S.H., Chu, P.H., Hsiao, P.Y., 2012. Data mining techniques and applications – a decade review from 2000 to 2011. Expert Systems with Applications 39, 11303–11311.

Likas, A., Vlassis, N., Verbeek, J.J., 2003. The global k-means clustering algorithm. Pattern Recognition 36, 451–461.

Marchant, J., 1996. Tracking of row structure in three crops using image analysis. Computers and Electronics in Agriculture 15, 161–179.

Montalvo, M., Pajares, G., Guerrero, J., Romeo, J., Guijarro, M., Ribeiro, A., Ruz, J., Cruz, J., 2012. Automatic detection of crop rows in maize fields with high weeds pressure. Expert Systems with Applications 39, 11889–11897.

Mostafa, M.M., 2013. More than words: Social networks– text mining for consumer brand sentiments. Expert Systems with Applications 40, 4241–4251.

Neumaier, A., 2004. Complete search in continuous global optimization and constraint satisfaction. Acta Numerica 13, 271–369.

Nievergelt, Y., 1994. Total least squares: state-of-the-art regression in numerical analysis. SIAM Review 36, 258–264.

Olsen, H.J., 1995. Determination of row position in small-grain crops by analysis of video images. Computers and Electronics in Agriculture 12, 147–162.

Pintér, J.D., 1996. Global Optimization in Action (Continuous and Lipschitz Optimization: Algorithms, Implementations and Applications). Kluwer Academic Publishers, Dordrecht.

Pla, F., Sanchiz, J., Marchant, J., Brivot, R., 1997. Building perspective models to guide a row crop navigation vehicle. Image and Vision Computing 15, 465–473.

Reyes, J., Morales-Esteban, A., Martínez-Álvarez, F., 2013. Neural networks to predict earthquakes in Chile. Applied Soft Computing 13, 1314–1328.

Rovira-Mas, F., Zhang, Q., Reid, J., Will, J.D., 2005. Hough-transform-based vision algorithm for crop row detection of an automated agriculture vehicle. Journal of Automobile Engineering 219, 999–1010.

Sabo, K., Scitovski, R., Vazler, I., 2013. One-dimensional center-based l_1-clustering method. Optimization Letters 7, 5–22.

Sabo, K., Scitovski, R., Vazler, I., Zekić-Sušac, M., 2011. Mathematical models of natural gas consumption. Energy Conversion and Management 52, 1721–1727.

Scitovski, R., Sabo, K., 2014. Analysis of the k-means algorithm in the case of data points occurring on the border of two or more clusters. Knowledge-Based Systems 57, 1–7.

Scitovski, R., Scitovski, S., 2013. A fast partitioning algorithm and its application to earthquake investigation. Computers & Geosciences 59, 124–131.

Scitovski, R., Šime Ungar, Jukić, D., 1998. Approximating surfaces by moving total least squares method. Applied Mathematics and Computation 93, 219–232.

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

Späth, H., 1983. Cluster-Formation und Analyse. R. Oldenburg Verlag, München.

Späth, H., 1981. Algorithm 48: afast algorithm for clusterwise linear regression. Computing 29, 175–181.

Teboulle, M., 2007. A unified continuous optimization framework for center-based clustering methods. Journal of Machine Learning Research 8, 65–102.

Tillett, N.D., Hague, T., 1999. Computer-vision-based hoe guidance for cereals - an initial trial. Journal of Agricultural Engineering Research 74, 225–236.

Vendramin, L., Campello, R.J.G.B., Hruschka, E.R., 2009. On the comparison of relative clustering validity criteria, in: Proceedings of the SIAM International Conference on Data Mining, SDM 2009, April 30 – May 2, 2009, Sparks, Nevada, USA, SIAM. pp. 733–744.

Yin, P.Y., 1998. Algorithms for straight line fitting using k-means. Pattern Recognition Letters 19, 31–41.