LiDAR-Based Online Navigation Algorithm For An Autonomous Agricultural Robot [*]

Dang Nguyen Thanh, Hung Le Van, Luy Nguyen Tan

Faculty of Electronic Technology, Industrial University of Ho Chi Minh City, Ho Chi Minh City, Vietnam
(e-mail: nguyenthanhdang@iuh.edu.vn, levanhung@iuh.edu.vn, nguyentanluy@iuh.edu.vn)

Abstract: In this paper, we propose a new approach for navigation of autonomous robots between crop rows in agriculture. This method is implemented by projecting 2D Light Detection and Ranging (LiDAR) data onto the robot's movement direction to perform one-dimensional Density-Based Spatial Clustering of Applications with Noise (DBSCAN). The mapping and localization (MAL) are obtained by combining the location of the virtual landmark from the DBSCAN with robot's position updated from the Particles filter. The results of this method map and estimate the robot's location simultaneously in one scan. Each robot's position depends on the LiDAR data information for the current scan and the previous scan. Data association to build global path is achieved by combining data from a number of consecutive scans and by the Kalman filter. The global trajectory, which is created by combining local positions, allows the robot to navigate autonomously in real time without having to go through the prior phases of collecting all of the data from the crop field. The article also performs FIR filter correction with different parameters to enhance the effectiveness of the proposed method.

Keywords: Agriculture, autonomous, control, laser, localization, mapping, mobile robot, navigation, sensing, vision, LiDAR.

1. INTRODUCTION

Autonomous navigation robots are widely applied in many fields such as military, medicine, space, traffic, etc. Robots hope to perform complex actions in indoor and outdoor environments (Cherubini, 2014; Lin, 2021; Da Mota, 2018; Zhang Y. a., 2020; Warren, 2019). Autonomous navigation robots in agriculture have also been an area of interest for current researches in many laboratories and companies (Zhang et al., 2018; García-Santillán et al., 2018; Malavazi et al., 2018; Rovira-Más, 2021; Durand-Petiteville, 2018). In the recent years, the agricultural economy has been developing rapidly due to the increasing demand of food. With the development of robot technology, agriculture is less dependent on the human labor. In particular, in the fields of extensive farming and fixed planning, using robots will bring efficiency in time and high economic efficiency because of their uniformity and timeliness. In the problem of taking care of crops that require chemicals such as pesticides and herbicides, which can affect human health, robots are necessary substitutes to ensure flexibility and safety. To fulfill the above requirements, it is necessary to have an autonomous navigation robot so that they can move between the crop rows on their own (García-Santillán et al., 2018; Meshram, 2021; Adamides, 2017). In order to perform autonomous navigation, the robot needs to locate itself in the environment and move safely without colliding with obstacles. The automatic navigation for agriculture robots is a combination of accuracy in sensor collection and data processing. Usually, that method incorporates several techniques to obtain the overall requirements. One of the most important tasks in autonomous navigation is to detect and extract features related to the crop rows. The traditional method uses image processing to detect crop rows in agriculture (García-Santillán et al., 2018; Zhang, et al., 2018). Zhang et al use a three-link phase image processing method, including: (i) Image segmentation (ii); Identification of starting points, and (ii) Crop row detection. Some techniques are used to detect agricultural crop rows, such as: Hough transform, Linear regression, Horizontal strips, Vanishing point, Filtering. In (García-Santillán et al., 2018), a method was proposed to identify both straight lines and curves by combining the Hough transform and Linear regression. In (Winterhalter, 2018), agriculture vehicles were equipped with sensors such as camera and Laser to detect crop rows. Hough transform and Random sample consensus (RANSAC) are the main methods to extract crop row features in that paper. The most common techniques in image processing are edge detection and object detection. These techniques are used to detect the desired feature, such as crop crows, weeds, and other object such as soil and stone. These methods, however, rely on collecting images from a camera, which is noise-sensitive and requires a high level of processing complexity, making them unsuitable for mobile robotic equipment.

A variety of other sensors such as Infrared, Radar, GPS, and IMU, which support autonomous navigation robots, were then put to the test (Xu Y. a., 2019; Liu, 2020; Bernardes, 2020). Nevertheless, the studies on these sensors also show that their inaccuracy and lack of flexibility with environmental conditions cause limits to their ability to extract the feature. For example, under a canopy of leaves, it is not suitable to rely on some sensors such as GPS or cameras due to loss of light and noise. With the limitations of the above types of sensors, LiDAR sensors have remarkable and outstanding

[*] This work was supported by the Industrial University of Ho Chi Minh City under Grant 23/HD-DHCN.

characteristics (Warren, 2019; Gallant, 2016; Zhang Y. a., 2020; Wang et al., 2021). The LiDAR sensor uses the light reflection mechanism to measure the range between the target and the sensor. Today, this sensor is very popular in the fields of autonomous navigation, such as UAVs, warehouse robots, etc. (Wang et al., 2021; Miao, 2021). LiDAR sensors have high durability, high accuracy, are less affected by environmental factors, have a large field of view, and are reasonably priced. LiDAR data is the point cloud coordinate of reflectors represented by the scanning angle and distance to the target from the sensor. Point cloud has high density if the number of scans is large for one position of scan. The features obtained from the environment are discrete points. The arrangement of discrete points will produce features such as lines, boundary objects or planes, points, etc. In agriculture, the reflective objects that LiDAR can detect are plants, canopies, stone, and bush. In particular, in addition to the required features, the LiDAR data captures a lot of noise from surrounding crop rows and plants distributed among the rows in the plant field, making it difficult to detect crop row features.

In (Taher, 2012), the omnidirectional vision is used for robot navigation, where the robot position is dynamically estimated. However, the operating space of the robot is limited in a small area of a Robocup competition field. In (María, 2012) a robot for navigation and inspect in oil wells is developed. The robot is equipped with sensors of sonar, encoder, IMU, and vision to measure environmental variables. However, the environment is absent from the obstacles; therefore, the LiDAR sensor probably not necessary. In (Tamas, 2010), authors applied LiDAR, computer vision, and Extended Kalman filter to detect and tracking people, where the robot position, which is relative to detection of the objects, is estimated. However, the environment is not considered a high noise density and a random distribution in the scanned areas, which is difficult to extract the feature of a straight line heading in a certain direction.

LiDAR point cloud data applied to autonomous navigation robot has attracted recent research (Choi et al., 2008; Meng, 2021; Yousif et al., 2015; Wang et al., 2021; Syaqur et al., 2018; Xu Y. a., 2019; Li T. a., 2021; Malavazi et al., 2018; Guan, 2021). With the development of computer vision, data processing techniques, and LiDAR sensor, many researchers have focused on studying LiDAR-based SLAM with two functions: localization and map building. This technique has been widely applied in autonomous navigation robots that allows mapping and localization simultaneously called SLAM. SLAM techniques using LiDAR sensors have been developed and used in many practical applications, such as AUVs, building maps for warehouse LOAM, etc. (Lin, 2021; Wang et al., 2021; Li T. a., 2021; Ramezani et al., 2020). Applications in autonomous navigation for complex feature extraction and object recognition using neural networks for LiDAR point cloud data have been studied in (Li, et al., 2020; Ramezani et al., 2020; Heinzler, 2020). Another algorithm relating to mapping is Gmapping, which uses the principle of Laser base SLAM for grid mapping (Esenkanova et al., 2013). This method requires making grid steps to create grid maps of environments. However, these methods are still complex in computation and are only useful for expensive hardware platforms such as navigation systems or secure robots, and are not suitable for mobile, low-cost platform applications such as appliance robots in clear feature environments. The problem, which relates to find the path for the robot to move between the rows of agricultural crops, was studied in (Malavazi et al., 2018). Random sample consensus (RANSAC) and Propose Expand and Re-estimate Labels (PEARL) algorithms can be used to reduce noise and find straight-line features (Malavazi et al., 2018). The work in (Malavazi et al., 2018) used the PEAL method to find features related to the crop rows in the LiDAR data. That work showed that the proposed method has improvements compared to the previous traditional methods such as Hough transform and RANSAC algorithm. Authors also proposed some other improved methods by using PEAL algorithm combined with FIR filter to control the robot moving precisely between two rows of plants. This method has proven to be more powerful than the previous methods in that there is no need to know any previous information about the crop rows. However, in the case of curved and broken rows of trees, we need a method that can be adaptively controlled to the change of the rows.

In line with the development of above-mentioned studies, we only utilize 2D LiDAR data and uses multi-scan technique to align the local path into the global path. This work is performed sequentially: scan, transform data, extract features, search optimal heading, combine data and enhance the results using Kalman filter, Fir filter. Mapping and localization (MAL) are achieved through these steps by finding a fixed virtual landmark result from DBSCAN (Ester et al., 1996) for 1D LiDAR data projected from 2D LiDAR data. The local position of the robot is defined by this virtual landmark and Particle filter. This step is called local feature extraction, and it is completed in a single scan position. The next step is to use a Kalman filter to combine the data and create a global path. This step is done by combining multiple scans in consecutively. The proposed method can perform autonomous navigation from starting point to ending point for crop rows with a straight or curved row distribution. The main **contribution** of this paper is the development of a mapping and locating approach and autonomous navigation algorithms for agricultural robots based on the density characteristic of LiDAR data.

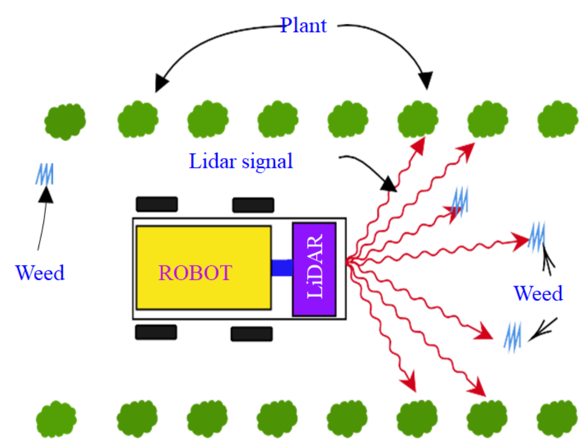

Fig. 1 LiDAR scan of two crop rows with noise distribution are weeds.

The paper is organized as follows: Section 1 gives an overview related to robot autonomous navigation used in agriculture and list the studies related to mapping, location and building global paths in recent times. Section 2 analyze methods considered in this paper related to feature extraction and building a global trajectory. Section 3 presents problem formulation. Section 4 gives details of the main steps of the algorithm. Section 5 is data and numerical simulation that give some discusses about this work and results, and the last section is conclusion and some suggestions about the future work.

2. ANALYSIS AND DESCRIPTION OF PROPOSED METHOD

The moving distance and orientation of the robot can be determined at a given angle (which is called heading pose) by analyzing and comparing features in two successive positions corresponding to two scans. Local features refer to the features that relate to a single scan at a specific robot's position. In agriculture, this feature relates to plant distribution in areas with a lot of other objects such as weeds, bush, stone, soil, etc. With the task of feature extraction, such as RANSAC algorithm (Fischler and Bolles, 1981) or Multimodel Fitting algorithm (Pham, 2014), can be applied into models that are straight-line distribution in the event that all data were collected. The algorithms can be used for determining the online location of robot if these models are segmented and then combined those segmented parts to get a global path of the robot. The disadvantage of this method is that it does not respond to curves or break line features, and the model is obtained based on the convergent probability of the algorithm. Clustering and DBSCAN (Ester et al., 1996) are two other methods that can be applied to LiDAR sensors with discrete point cloud data. This method proves to be very effective when data points are density distributions in 2D, or 3D space for the models with continuous density distribution. However, they are only suitable for removing noises from desired features and need to be incorporated with other methods to detect features. For example, Kmean clustering (Xiong, 2009) can be used to cluster data points into two parts. This part is used to create the reference pose for the robot if the point cloud data is symmetric in the robot's heading. Data clustering methods can be considered as preprocessing methods, which are applied to remove noise for outdoor environments.

With the characteristics of the data related to the crop rows, where the objects are small in size and discretely distributed, the models are also distributed in many directions. This environment also has a high noise density and a random distribution in the scanned areas. In such an environment, it is very difficult to extract the feature of a straight line heading in a certain direction. Point cloud data collected from the agriculture environment with discrete features has potential for complexity like data collected from the road by autonomous vehicles. Therefore, applying the LiDAR SLAM method to agriculture is not necessary because it requires two stages: The first step is building the map and the second step is defining the global path. That method has only a small adjustment on the position of the robot and the map. According to the above analysis, it is appropriate to apply clustering methods to discrete data points as a preprocessing step to determine local features. This paper proposes a method to extract local features by using DBSCAN on one-dimensional LiDAR data and determine the local robot's position in one scan. DBSCAN (Ester et al., 1996) is a density-based clustering method that can be used to eliminate noise and determine global models in offline processing. Two important parameters in DBSCAN are (i) the distance of points distributed into a data cluster and (ii) the minimum number of points that satisfy the clustering condition in (i). The analysis of the feature extraction of the local and global models described above in this work concludes that: depending on the data and environment, these methods have advantages and limitations. With the environment is field of plant, to find the global model using LiDAR data in crop rows we combined many local features in consecutive scans. Thus, this work does not need to perform SLAM to find the map first and then build the global path. Instead, we apply the one-dimensional DBSCAN to find local features (virtual landmarks). Based on these virtual landmarks, we determine the local position of the robot in one scan of the LiDAR sensor. The approach merges data points from 2D into 1D in this way allowing it to construct local feature information without losing generality. The result of DBSCAN is one feature point, which is called a "virtual landmark of local feature", instead of a line feature. With a given rule about the relative position of a virtual landmark and a robot pose, that local feature can be combined to build a global path for the robot from the starting point to the ending point.

3. PROBLEM FORMULATION

The robot moving between the crop rows and receiving LiDAR signal, which is reflected by the plants and weeds, is depicted in Fig. 1. The configuration of the plant field and the truth path that the robot must follow are shown in Fig. 2. The configuration of the plant field is described as follows:

- The crop rows are distributed in a straight or curved line.
- Plants can be distributed vertically or horizontally.
- There is a distribution of noise that mainly comes from weeds.
- LIDAR data contains the desired features that are Laser signals reflected by trunk of plants or by small canopy.
- Crops can disperse according to a given deviation.
- The ground surface of the field is assumed to be flat.
- The plants need to be avoided as obstacles.

The requirement is that the robot start from the starting point, searching for the direction of the destination and facing the crop rows, scanning and detecting features, mapping and locating, and building a path to perform the necessary operations such as fertilizing, spraying, etc. In that scenario, the environment is the field of plants. Each plant has the same role as an obstacle that needs to be avoid. With a scan at a certain position, the LiDAR sensor emits a light signal and receives the reflected signal to measure the range. Since the plants are arranged in a straight line, even in any direction, the adjacent row is collected, which is treated as unnecessary information like noise and cause difficult to define a required model. Depending on the age of the plants, the diameter of the trunk can be large or small. Plants near the scanning position have a higher density of collection points than those with a farther position.

In order to build global path, the following basic assumptions be satisfied:

- The robot must be located through a series of scans. The robot's position is determined by using extracted feature and building a local model to form a virtual landmark.
- The robot can find a path that avoids colliding with the plants.
- The robot can recognize the end of each line to perform a turn.

In reality, the environment contains imperfect features, and due to the sensor noise, the information of robot's position and the map often has potential errors. The result of the global path depends on the information of the robot's position in a single scan, so it is very important to determine the exact location in one scan at a certain robot's position. The combination of consecutive local positions will create a global path that the robot can make autonomous navigation in this way.

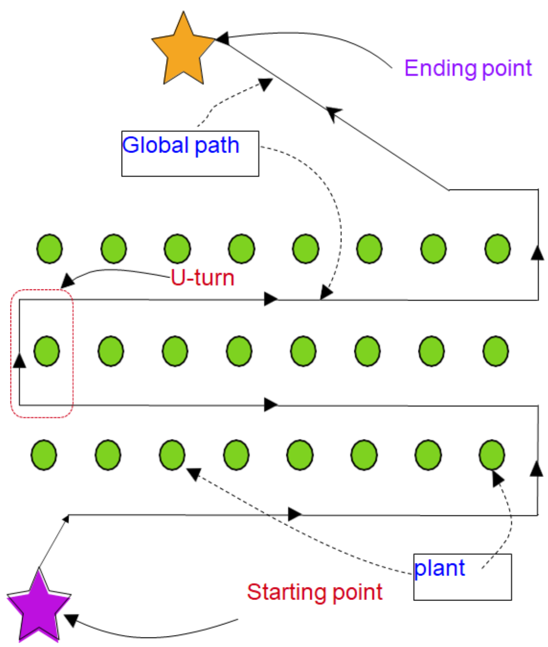

Fig. 2. Configure of plant field and global truth path.

The useful of feature extraction method used in this section is DBSCAN. This method is applied to cluster and classify data from noise, and it is used to identify features distributed in data points. In this paper, we utilize one-dimensional DBSCAN to find features of crop row. This method aims to find the exact positions of virtual landmarks that have certain characteristics in distribution of data points. Firstly, LiDAR point cloud data are projected onto the robot's heading in one scan as shown in Fig. 3. The projected data is then performed DBSCAN with the relevant parameter (these parameters are the distance between points and the minimum number of points in a cluster). Next, when the data is clustered, the distance $d(\theta_t)$ that the cluster forms on the projecting direction is calculated ($d(\theta_t)$ is annotated in Fig. 3). Finally, we find out the angle of heading pose that formed the smallest distance $\theta_t^{opt} = \operatorname{argmin}(d(\theta_t))$, where $\theta_t^{opt} \in \left[\theta_{t-1}^{opt} - \Phi; \theta_{t-1}^{opt} + \Phi\right]$, $\Phi \le \frac{\pi}{2}$. Φ is range of angle to search. Given $lm(x,y)$ is the coordinate of the virtual landmark, which needs to be determined, and DB is the set of points on the estimated angle satisfying the following

$$lm(x,y) = \left(\frac{\max_x(DB) - \min_x(DB)}{2}, \frac{\max_y(DB) - \min_y(DB)}{2}\right) \tag{1}$$

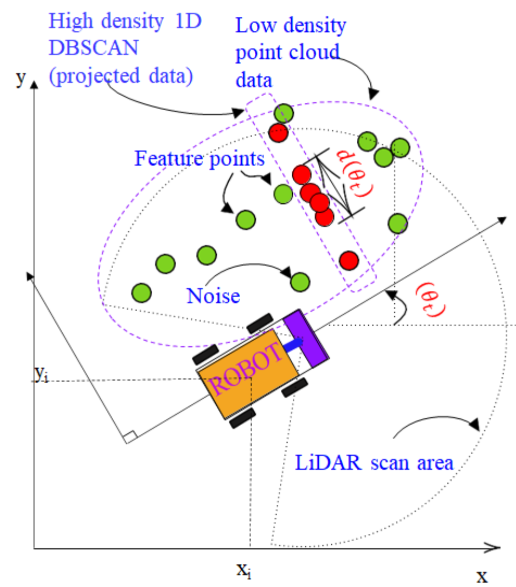

Fig. 3 Projected data for 1D DBSCAN (red points). With the pose heading angle θ_t, we have the $d(\theta_t)$ is distance spanned by points clustered by 1D DBSCAN.

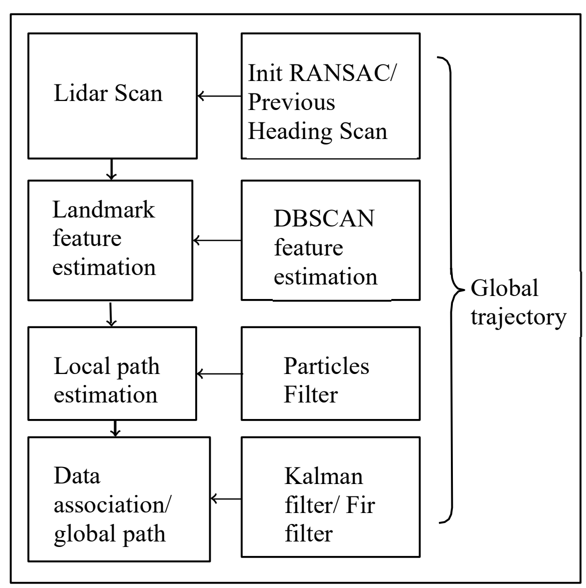

Fig. 4. The general flow diagram for implementation of proposed method.

If the line of plant is curved, it is considered as a set of straight lines in a small area. The discrete feature points from the 2D LIDAR point cloud that are distributed in a straight line will be aggregated to form a high-density one-dimensional cluster. Using this method, obstacles with a straight-line distribution, such as lines of plant, will become some points with piled-up distribution in the moving direction, making them easily distinguishable from other locations as noise points. The optimization of the pose angle is carried out by searching on the range of angles that its center is angles formed by projecting direction. MAL is performed in this scan by using the obtained virtual landmark and particles filter to determine the position of the robot. For data association between scans to build the global path, a Kalman filter is used to process feature parameters extracted from the current scan and the previous scan. The general flow diagram for implementation of the proposed method is shown in Fig. 4, and the main algorithm of this method are the combination of MAL and filters which is explained as following steps:

1) Find the initial direction for the robot using RANSAC.
2) Scan and project data on the direction of the robot moving.
3) Perform DBSCAN on projected data and extract features from the collected data. If the data satisfied the end-of-row condition, make U-turn and return to step 2. If the end-of-field of plant condition is satisfied, switch to step 8.
4) Find the optimal DBSCAN distance (minimum).
5) Locate a virtual landmak based on step 4.
6) Update the position of the robot with Particles filter based on a given rule so that when performing step 7 to form a global path for the robot.
7) Combine previous data and form the global path using Kalman filter, be smooth the path by FIR filter and go back to step 2.
8) Go to the ending position.

4. THE MAIN STEPS OF THE ALGORITHM

In this section we present the main steps of the proposed method to implement navigation for the robot. The three main points in this section will follow from finding local feature by DBSCAN, updating robot position by Particle filter, and building global path with Kalman filter.

4.1 Find the local feature.

In order to reach the ending point, one vector is established with a given starting point and ending point. In step 1 of section 3, the robot moves toward the ending point in the direction of this vector to search the first row of the plant field in which the plant as the obstacle needs to be avoided. This is done by using RANSAC iteration to obtain the initial heading and initial robot's position, which are used to determine the robot pose in the next step. This step is detailed in **Algorithm 1**. Local features will be performed from step 2 to step 6 in section 3. Extracting line features and locating virtual landmarks for crop rows are achieved via DBSCAN in one scan. The history of the robot's position $S_0, S_{t-n-1}, \dots S_{t-1}, S_t$ in the global coordinate system is used to extract the local feature map. The state t of the robot in scan is represented as following:

$$\begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{bmatrix} + \begin{bmatrix} v_t \Delta t \cos(\theta_{t-1}) \\ v_t \Delta t \sin(\theta_{t-1}) \\ \omega_t \Delta t \end{bmatrix} \tag{2}$$

where θ_t is the estimated angle at time t. The angular velocity ω_t is chosen by zero since θ_t is predicted equal to the previous angle θ_{t-1}, and it will be updated by from DBSCAN.

Algorithm 1 RANSAC algorithm to find the initial direction angle and pose

Global angle (*startpoint, endpoint*);
Line (*angle, robotpose*)
Numinline (*Data, Line*);
Input : {(*startpoint, endpoint, inlier*)}
Output: *angle init, pose init*
while *num inlier < inlier* **do**
    Update robotpose[i] from startpoint to endpoint with stepsize=20
    Data[i]=DataSwep from LiDAR
    for *b = 0 to K* **do**
        angle[b]= $\frac{3.14}{K} * b$
        d=Line(angle[b],robotpose[i]);
        Numinline=Numinline(Data[i],d);
        if Numinline>inlier **then**
            return (robotpose[i], angle[b]) ; break ;
        end
    end
    i+=1;
end

The robot's step size $v_t \Delta t$ is fixed and thus constant. The coordinates x_t, y_t are the predicted position that will be updated after performing the Particles filter.

Define dx, dy as the travel distance between two scans in the x and y directions:

$$\begin{aligned} dx &= x_t - x_{t-1} = v\Delta t \cos(\theta_{t-1}) \\ dy &= y_t - y_{t-1} = v\Delta t \sin(\theta_{t-1}) \end{aligned} \tag{3}$$

Let $\vec{u}_1 = (dx, dy)$ be the vector in the direction of motion of the robot. The projecting direction $\vec{u}_2$ of the data is perpendicular to $\vec{u}_1$. With $\vec{u}_2 = (-dy, dx) = (u_{21}, u_{22})$, Let $\varphi_1 = \langle \vec{u}_1, \vec{i}_x \rangle$ and $\varphi_2 = \langle \vec{u}_2, \vec{i}_x \rangle$, since $\vec{u}_2 \perp \vec{u}_1$ therefore $\varphi_2 = \frac{\pi}{2} + \varphi_1$. To optimize value DBSCAN according to projecting direction, the angle to search must be in range:

$$\theta_i^t = \{\beta_i \mid \beta_i \in [\varphi_1 - \Phi, \varphi_1 + \Phi], i = 0,1,\dots,k\} \tag{4}$$

with the step sampling $\Delta\theta = \frac{\pi}{k}$, $\Phi \le \frac{\pi}{2}$. For a certain θ_i^t, we find the corresponding unit vector $\vec{v}_{i1}^t = (v_{i11}, v_{i12}) | \theta_{i1}^t = \langle \vec{i}_x, \vec{v}_{i1}^t \rangle$, θ_i^t and the perpendicular vector $\vec{v}_{i2}^{t} = (-v_{i12}, v_{i11}) | \vec{v}_{i2}^{t} \perp \vec{v}_{i1}^{t}$ therefore $\theta_{i2}^{t} = \langle \vec{\imath}_x, \vec{v}_{i2}^{t} \rangle = \theta_{i1}^{t} + \frac{\pi}{2}$. Assume that one scan position has n points $S_t = \{s_1, s_2, \dots s_n\}$, each point corresponding to a vector point. Thus, data points are a set of vector points $\tilde{S} = \{\vec{s}_1, \vec{s}_2, \dots \vec{s}_n\}$. The projection of point cloud on $\vec{v}_{i2}^{t}$ will be $s_{2i} = \vec{s}_i . \vec{v}_{i2}^{t}$, and the projection of robot's position on $\vec{v}_{i2}^{t}$ becomes $s^t = x_t \Omega_{11} + y_t \Omega_{12}$. Let $\Omega_{11} = \cos(\theta_{i1}^{t})$, $\Omega_{12} = \sin(\theta_{i1}^{t})$, $\Omega_{21} = \cos(\theta_{i2}^{t})$, $\Omega_{22} = \sin(\theta_{i2}^{t})$. The y coordinates of the projection are now written as:

$$\begin{bmatrix} \tilde{s}_{21} \\ \tilde{s}_{22} \\ \vdots \\ \tilde{s}_{2n} \end{bmatrix} = \begin{bmatrix} \Omega_{22} & 0 & \cdots & 0 \\ 0 & \Omega_{22} & \cdots & 0 \\ \vdots & & \ddots & \\ 0 & 0 & \dots & \Omega_{22} \end{bmatrix} \begin{bmatrix} s_{21} \\ s_{22} \\ \vdots \\ s_{2n} \end{bmatrix} + \begin{bmatrix} \Omega_{12} \\ \Omega_{12} \\ \vdots \\ \Omega_{12} \end{bmatrix} s^t \tag{5}$$

Then we have data on the axis $(\vec{v}_{i1}^{t}, \vec{v}_{i2}^{t})$ in form $\{\tilde{s}_1, \tilde{s}_2, .. \tilde{s}_n\} = \{(\tilde{s}_{21}, \Omega_{11} s^t), (\tilde{s}_{22}, \Omega_{11} s^t) \dots (\tilde{s}_{2n}, \Omega_{11} s^t)\}$.

The next step is to perform DBSCAN for the projected data points $\{\tilde{s}_1, \tilde{s}_2, .. \tilde{s}_n\}$. In fact, we only find the one-dimensional DBSCAN for the values that are the y coordinates of the projected points because the x coordinates are the same value in new coordinate system $(\vec{v}_{i1}^{t}, \vec{v}_{i2}^{t})$, which means the set of points to find DBSCAN will be $DT_{\theta_{i1}^{t}} = \{\tilde{s}_{21}, \tilde{s}_{22}, .. \tilde{s}_{2n}\}$. The parameters for DBSCAN are $min\_cluster\_size$ and $min\_samples$, which are chosen based on the distribution of point cloud data heuristically. Let $FT_{\theta_{i1}^{t}} = \{L_1, L_2, \dots, L_m\}$ be the set of points with the maximum density as the features have been extracted, where m is the number of features, and $FT_{\theta_{i1}^{t}} \subset DT_{\theta_{i1}^{t}}$, $m < n$. The length of segment spanned by points projected in the direction $\vec{v}_{i2}^{t}$ is:

$$d\left(\theta_{i1}^{t}\right) = \sqrt{(A)^2 + (B)^2} \tag{6}$$

where $A = \max_x(DB_{\theta_{i1}^{t}}) - \min_x(DB_{\theta_{i1}^{t}})$, and

$B = \max_y(DB_{\theta_{i1}^{t}}) - \min_y(DB_{\theta_{i1}^{t}})$. The optimal angle θ_t^{opt} can be calculated using range angles in (4) as follows:

$$\theta_t^{opt} = \operatorname{argmin}\left(d\left(\theta_{i1}^{t}\right)\right) \tag{7}$$

As a result, the estimated position of the virtual landmark is determined as the mean point of $d(\theta_t^{opt})$. Thus, the virtual landmark coordinates are:

$$lm_t(x, y) = \left(\frac{A}{2}, \frac{B}{2}\right) \tag{8}$$

If the result of DBSCAN processing has no features, the number of point cloud will be less than a certain threshold, which means it is the end of the crop rows. The U-turn will occur in step 3 of the main algorithm in section 3.

4.2 Robot's local position updated by Particles filte.

The initial position of the robot is specified in this model by adding a fix distance b to a baseline (from the virtual landmark) which is annotated in Fig. 5. The robot's exact position will be updated and refined using a Particles filter. Using this position, we can calculate the next estimated angle and obtain important information for the next scan. Mapping and Location (MAL) refers to the process of determining the map and location in one scan, which is accomplished in two phases: DBSCAN and Particles filter. Base on the virtual landmark obtained from DBSCAN, we use the Particles filter (Nie, 2020; Du, 2015) to update the robot's location. In this case, the virtual landmark is considered as an observed landmark. The motion equations and observed landmark of the system are shown as:

$$\begin{cases} p_t = f(p_{t-1}, u_{t-1}, z_t(lm_t), \delta_t) \\ lm_t = DBSCAN(p_{t-1}) \end{cases} \tag{9}$$

with $z_t = \{lm_1, \dots lm_{t-1}, lm_t\}$ is the history of the observed landmarks in consecutive data scan of the robot, δ_t is noise measurement, u_t is the input of system motion and $p_t = \begin{bmatrix} x_t \\ y_t \end{bmatrix}$ is the location of robot at time t. Using the Bayes' rule, the probability of observed landmark and robot's location can be decomposed into:

$$P(lm_t, x_t \mid x_{t-1}, z_t, u_{t-1}) = P(lm_t \mid x_{t-1}, z_{t-1}) P(x_t \mid x_{t-1}, u_{t-1}) \tag{10}$$

where $P(p_t \mid p_{t-1}, u_{t-1})$ is the positioning problem. The current position p_t is predicted using the previous estimated angle and previous position p_{t-1} which is presented by:

$$\begin{bmatrix} x_t \\ y_t \\ \theta_t^{opt} \end{bmatrix} = \begin{bmatrix} 1 & 0 & robot\_step.\cos\left(\theta_{t-1}^{opt}\right) \\ 0 & 1 & robot\_step.\sin\left(\theta_{t-1}^{opt}\right) \\ 0 & 0 & \theta_{t-1}^{opt} \end{bmatrix} \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ 1 \end{bmatrix} \tag{11}$$

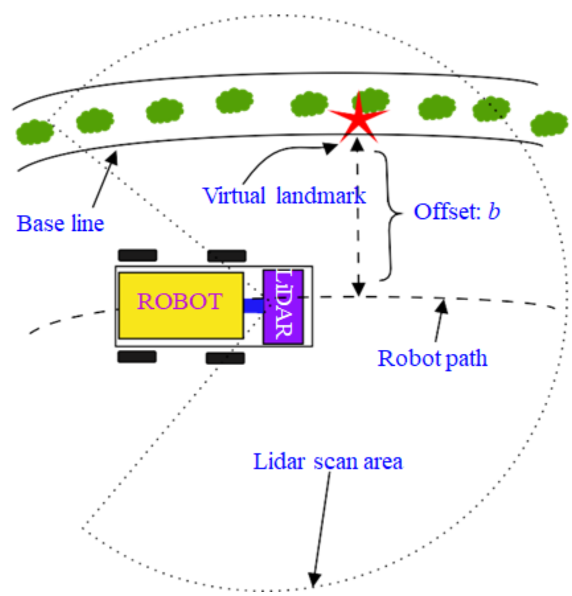

Fig. 5. Online LiDAR one scan and local path. A virtual landmark is the base to determine local position of Robot by a fix distance (offset).

where $robot\_step$ is the distance between two scan positions defined as $robot_step = \sqrt{dx_{t-1}^{\ 2} + dy_{t-1}^{\ 2}}$, θ_{t-1}^{opt} is the optimal angle at time $t-1$ in (7) and thus the transition model can be defined as:

$$u_t = \begin{bmatrix} 1 & 0 & robot\_step.\cos\left(angle_{t-1}^{opt}\right) \\ 0 & 1 & robot\_step.\sin\left(angle_{t-1}^{opt}\right) \\ 0 & 0 & angle_{t-1}^{opt} \end{bmatrix} \quad (12)$$

The robot position is now approximated as:

$$p_t \sim N(u_t p_{t-1} + n_t, R) \quad (13)$$

where N is a Gaussian distribution function with mean vector $u_t p_{t-1} + n_t$ and variance matrix R.

$P(lm_t \mid p_{t-1}, z_{t-1})$ is the mapping problem in which lm_t is estimated by DBSCAN via (8). The probability of the landmark is conditional independent with the robot's position. In this work the map is lm_t providing a basic landmark for updating the robot's position. According to Particles filter applied in (Du, 2015) we derived:

$$P(x_t, lm_t) = \sum_{i=1}^{N_s} \omega_t^i \delta(p_t - p_t^i) \quad (14)$$

where N_s denotes the number of samples in the Particles filter, w_i denotes the weight assigned to particle i, δ is Dirac Delta function.

Let zlm_t be the distance from p_t to observed landmark lm_t, the probability of observed landmark from predicted position of robot p_t is $p(lm_t \mid p_t) = N(zlm_t; Hp_t, Q)$. Thus, the weight for each particle in (14) is given by (Du, 2015):

$$w_t^j \propto w_{t-1}^j p(lm_t \mid p_t) \quad (15)$$

where N is Gaussian distribution with the mean vector equal to Hp_t and variance matrix equal to Q, H denotes measurement error model. The expression in (14) is used to estimate the final position of robot in one position scan. So far, the steps 2 through 6 in section 3 are obtained.

4.3 Data association by Kalman filter

The local position obtained from particle filter p_t is considered as a predicted position of the global path. This position needs to be adjusted by Kalman filter. The following expression now establishes a constraint between the observed and predicted values:

$$O_t = Hp_t + n_t \quad (16)$$

where:

- H is the observation model, which maps the predicted position p_t into the observed position O_t
- n_t is the observation noise, which is assumed to be zero mean Gaussian white noise with covariance N_t.

The predicted states of the robot position are defined as:

$$\bar{p}_t = A\bar{p}_{t-1} + B.a + \delta_t$$
$$\Leftrightarrow \begin{bmatrix} p_t \\ v_t \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p_{t-1} \\ v_{t-1} \end{bmatrix} + B.a + \delta_t \quad (17)$$

where:

- $A = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}$ is a transition model used to predict new positions for the robot, $v_t = \frac{x_t - x_{t-1}}{\Delta t}$, which yields the covariance matrix $\bar{P}_t = A_t P_{t-1} A_t^T + Q_t$, $P_{t=0}$ is the variance matrix for the estimated initial position. $P = \begin{bmatrix} \Sigma_{p_t p_t} & \Sigma_{p_t v_t} \\ \Sigma_{v_t p_t} & \Sigma_{v_t v_t} \end{bmatrix}$ where p_t is the position estimation error and v is the velocity estimation error relative to the movement of the robot. $\bar{p}_t = \begin{bmatrix} p_t \\ v \end{bmatrix}$ is the robot states predicted based on the previous states $\bar{p}_{t-1} = \begin{bmatrix} p_{t-1} \\ v_{t-1} \end{bmatrix}$. Data association is obtained in this manner through the consecutive scan positions.
- δ_t is the measurement noise, which is considered a zero-mean multivariate normal distribution with covariance Q_t.
- B is the transition model to control acceleration a. With a nearly linear model we can approximate the acceleration $a = 0$ and thus $B.a = 0$

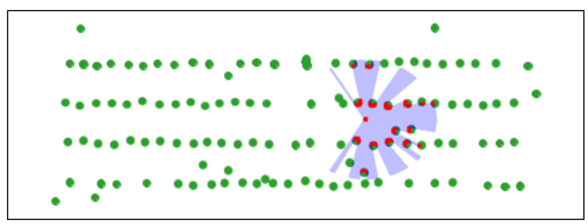

Fig. 6. LiDAR scan area with blue Laser rays and data point cloud with red point.

In this work, the robot's velocity is assumed to be constant $v_t = v_{t-1} = v$. Δ_t is period of time that the robot moves with velocity v in the distance $robot\_step$. Finally, the Kalman filter (Xu W. a., 2021) parameters used to update the robot's location are written as:

$$K_t = \bar{P}_t H_t^T (H_t \bar{P}_t H_t^T + R_t)^{-1} \quad (18)$$

$$P_t = \left(I - K_t H_t\right) \bar{P}_t \quad (19)$$

$$\hat{p}_t = \bar{p}_t + K_t\left(z_t - H_t \bar{p}_t\right) \quad (20)$$

with $\hat{p}_t$ is the robot's position updated by Kalman filter. The global path for the robot is obtained by incorporating the updated position in consecutive scan position. The step 7 of the main algorithm in section 3 will be completed here.

4.4 Processing of the U-turn and FIR filter for smooth global path

Processing of the U-turn at the end of crop rows and reach the end point (step 8 in section 3) are presented in this section. The end of each row is inferred by the previous information and the changes of data information in the last position. Each scan position, the number point cloud and local feature are calculated. If the number of data points is less than a specified threshold and the features is not found, the U-turn will take place and the robot pose will be infer from previous crop rows. If the local feature is not found in consecutive scan position, the robot will move to the ending point.

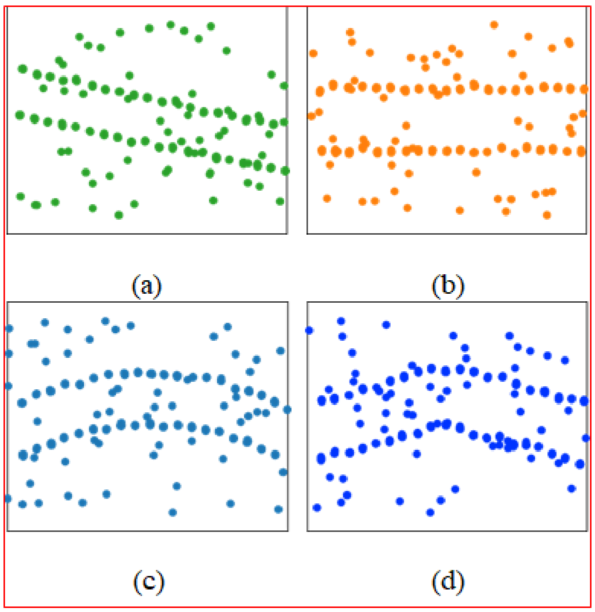

Fig. 7 The configuration of a plant field that yields data point cloud in certain two rows: a) Two rows feature in tilt angle, b) Two rows feature in horizontal direction, c) Two rows feature incurve distribution and d) Features in obtuse angle.

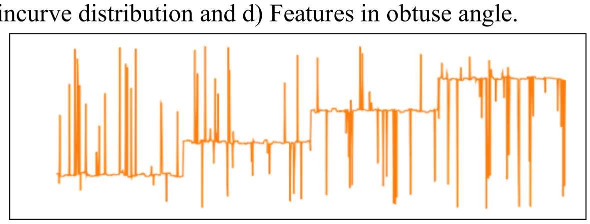

Fig. 8. Noise distribution in feature data.

A finite impulse response filter is known as a FIR filter. FIR filters are used in digital signal processing to remove specific frequency bands from input signals. In determining the path of robot, the global path predicted by the Particles filter may be rough and it needs to be smooth by the FIR filter (Malavazi et al., 2018; Xu Y. a., 2019). This article uses percentage coefficients to adjust the smoothness of the global path according to the nonlinearity of the path. Table 1 lists the coefficients that were heuristically chosen based on the best simulation results.

5. NUMERICAL SIMULATION

The parameters of a real-world plant field are used to generate simulation data. These parameters include the width of the crop rows, the distance between the plants, rows bias and the noise threshold as shown in Table 2. The weeds that grow among the plant's rows are considered noise in the collected data. Because of the complex characteristics of the agricultural environment, the yield of the plant in this work is assumed to be flat and parallel to the sweep of LiDAR. This means that LiDAR data is only received from the signal reflected from the plant trunk, not from the ground. In the general case, the global data is collected over a field of plants. We also assume that the space between each row of plants is enough for the robot to take a U-turn to continue scanning another row of plants. The data points have a discrimination between the model points and the noise points with a given ratio. Fig. 8 shows noise distribution in the desired feature data. The main parameters influencing the robot's trajectory shape include the number of point cloud reflected by the trunk (which depend on speed of sweep of LiDAR sensor). The ratio of the received noise point to the desired feature's data point, plant spacing, the width of the rows, and the bias of the plant (the plant is not aligned). The standard unit for these spacing parameters is row spacing that is within the range of LiDAR scan (unit in meters), and other values are percentages of this value. The details of parameters in the LiDAR data are listed in Table 2. Point cloud data is simulated according to the real environment which is described in Fig. 6 in four scenarios as shown in Fig. 7.

Table 1. Parameters of fir filter.

Plant rows Configuration	FIR coefficients		
	coefficient 1	coefficient 2	coefficient 3
Straight line	0.5	0.3	0.2
Broken line	0.8	0.1	0.1
Curve line	0.7	0.2	0.1

Table 2. Data parameter simulation.

Heading angle	**Parameters related to rows width**			
	Plant spacing	***The row bias***	***Width of row***	***Noise Threshold***
0.1 rad	30%	20%	100%	0.5
0 rad	40%	20%	100%	0.5

Simulation results for each different scenario are shown in Fig. 9-12. The results in Fig. 9 and Fig. 12 are configurations of the plant field detected by MAL and by filters with the rows distributed in a straight line. The processing algorithm determines the position of the robot at the center of two rows of plants. Although the data contains a lot of noise, all of the lines obtained are smooth, and the line without the filter is more distorted than the filtered line. The data points have a distinct feature and a symmetric distribution at the end of each row, allowing the algorithm to process U-turns with good results. As shown in Fig. 12, there is an initial disturbance when the row direction is angled with respect to the approach direction of the robot. The advantages of the algorithm are shown in Fig. 10 and Fig. 11 when rows of plant have broken and curved shape with noise.

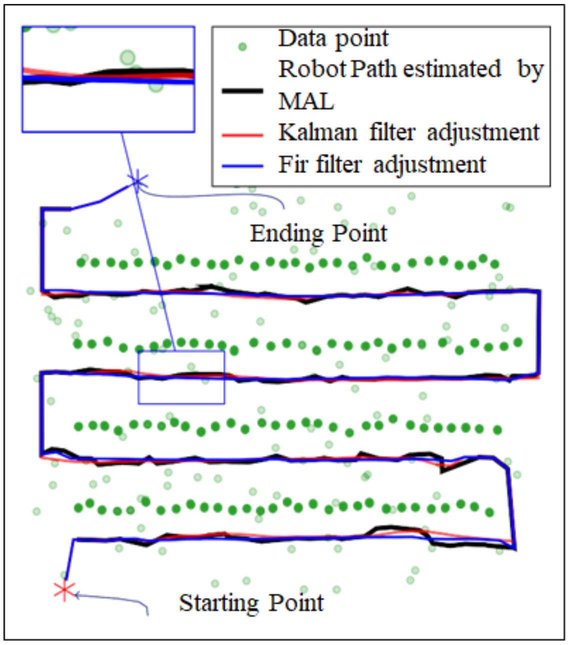

Fig. 9. Global path for straight line, pose heading angle=0.

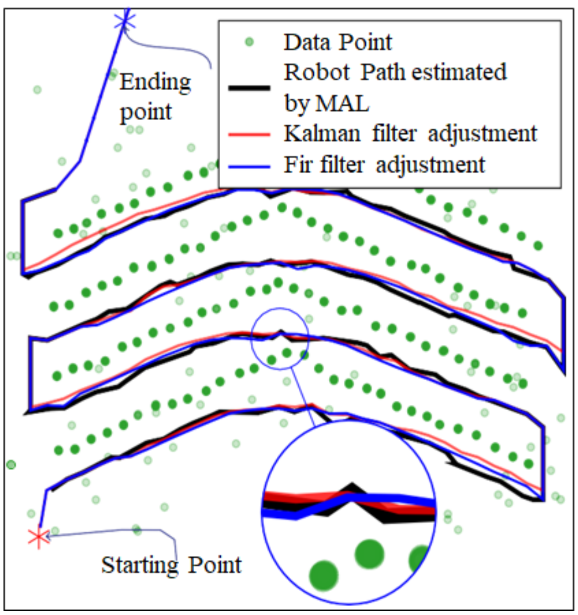

Fig. 10. Global path for broken shape in plant row.

The results show that the processing algorithm still finds the correct path that does not collide with the rows of plant. In these situations, the Kalman filter can play a more vital role, and the separated path is clearly visible. Without this filter, the robot's scanning direction shifts and drifts, causing data collection to go in the wrong direction. The algorithm, however, only handles inclinations of less than 0.4 rad. Above this value the Kalman filter does not respond to the parameter used previously so it will cause errors in the robot's pose. The FIR filter has the effect of smoothing out rough lines. Parameters are analyzed and heuristically selected to be compatible with each row distribution feature.

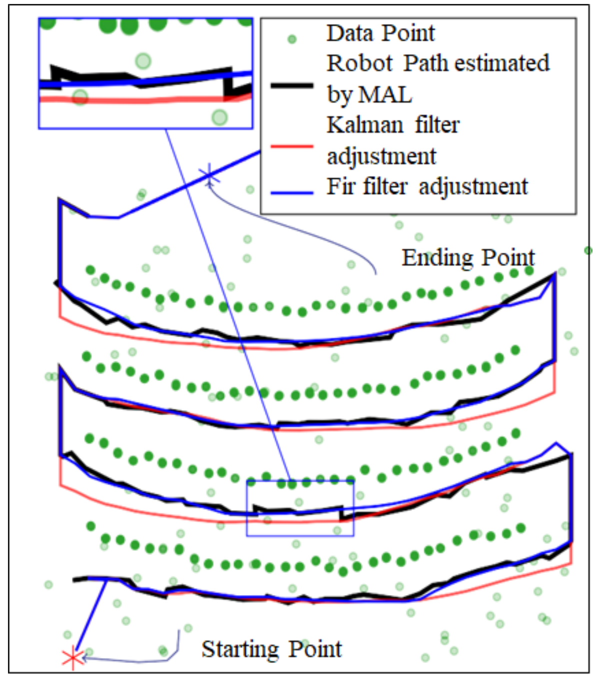

Fig. 11. The zoom region reveals that one processing has a wrong angle pose, but with the role of the Kalman filter, the angle pose is corrected, and the right path is maintained.

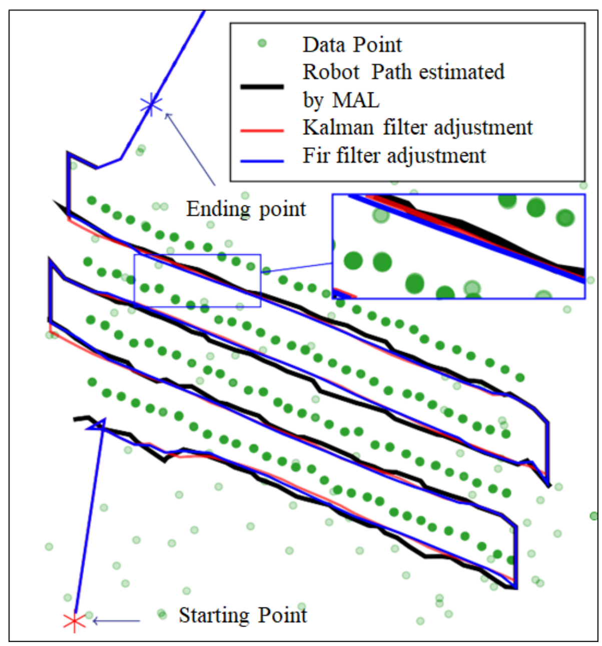

Fig. 12 The global path for the tilt angle between the plant row and the pose heading.

In various processing scenarios, the parameters listed in Table 1 provide the best results.

6. CONCLUSION

A novel approach for locating and determining the path for robots has been implemented and utilized in the article. The results show that DBSCAN combined with Particles filter is a good solution in noisy environments with various crop row features to detect. The global path estimated by the Kalman filter with smooth lines from various configurations of the plant field show that the proposed algorithm can be used in practice.

The method has the advantage of allowing online local path processing in a single scan at the robot's position without having to grid the environment. The future work will focus on expanding this method to detect more features and applying it to different environments such as roads, warehouses etc.

REFERENCES

Adamides, G. a. (2017). Design and development of a semi-autonomous agricultural vineyard sprayer: Human-robot interaction aspects. *Journal of Field Robotics, 34*, 1407-1426.

Bernardes, E. a. (2020). A Three-Photo-Detector Optical Sensor Accurately Localizes a Mobile Robot Indoors by Using Two Infrared Light-Emitting Diodes. *IEEE Access, 8*, 87490-87503.

Cherubini, A. a. (2014). Autonomous Visual Navigation and Laser-Based Moving Obstacle Avoidance. *IEEE Transactions on Intelligent Transportation Systems, 15*, 2101-2110.

Choi, Y.-H., Lee, T.-K., & Oh, S.-Y. (2008). A line feature based SLAM with low grade range sensors using geometric constraints and active exploration for mobile robot. *Autonomous Robots, 24*, 13-27.

Da Mota, F. A. (2018). Localization and Navigation for Autonomous Mobile Robots Using Petri Nets in Indoor Environments. *IEEE Access, 6*, 31665-31676.

Du, G. a. (2015). A Markerless Human–Robot Interface Using Particle Filter and Kalman Filter for Dual Robots. *IEEE Transactions on Industrial Electronics, 3*, 2257-2264.

Durand-Petiteville, A. a. (2018). Tree Detection With Low-Cost Three-Dimensional Sensors for Autonomous Navigation in Orchards. *IEEE Robotics and Automation Letters, 3*, 3876-3883.

Esenkanova, J., Ilhan, H., & Yavuz, S. (2013). Pre- mapping system with single laser sensor based on gmap ping algorithm. *International Journal of Electrical Energy, 1*, 97–101.

Ester, M., Kriegel, H., Sander, J., & Xu, X. (1996). Adensity-based algorithm for discovering clusters in large spatial databases with noise. *In kdd, 96*, 226-231.

Fischler, M.A. and Bolles, R.C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM, 24(6)*, 381–395.

Gallant, M. J. (2016). Two-Dimensional Axis Mapping Using LiDAR. *IEEE Transactions on Robotics, 32*, 150-160.

García-Santillán, I., Guerrero, J., & Montalvo, M. (2018). Curved and straight crop row detection by accumulation of green pixels from images in maize fields. *Precision Agric 19, 19*, 18-41.

Guan, W. a. (2021). Robot Localization and Navigation Using Visible Light Positioning and SLAM Fusion. *Journal of Lightwave Technology, 39*, 7040-7051.

Heinzler, R. a. (2020). CNN-Based Lidar Point Cloud De-Noising in Adverse Weather. *IEEE Robotics and Automation Letters, 5*, 2514-2521.

Li, T. a. (2021). PPP-LOAM: PPP/LiDAR Loosely Coupled SLAM With Accurate Covariance Estimation and Robust RAIM in Urban Canyon Environment. *IEEE Sensors Journal, 21*, 6660-6671.

Li, Y., Ma, L., Zhong, Z., Liu, F., Chapman, M., Cao, D., & Li, J. (2020). Deep Learning for LiDAR Point Clouds in Autonomous Driving: A Review. *IEEE Transactions on Neural Networks and Learning Systems, 32*, 1-21.

Lin, H.-Y. a.-Z. (2021). Autonomous Quadrotor Navigation With Vision Based Obstacle Avoidance and Path Planning. *IEEE Access, 8*, 102450-102459.

Liu, W. a. (2020). A Novel Multifeature Based On-Site Calibration Method for LiDAR-IMU System. *IEEE Transactions on Industrial Electronics, 67*, 9851-9861.

Malavazi, F., Guyonneau, R., Fasquel, J., Lagrange, S., & Mercier, F. (2018). Lidar-only based navigation algorithm for an autonomous agricultural robot. *Computers and electronics in agriculture, 154*, 71-79.

María, U., et al. (2012). Development of a novel autonomous robot for navigation and inspect in oil wells. *Journal of Control Engineering and Applied Informatics*, 14(3), 9-14.

Meng, J. a. (2021). Efficient and Reliable LiDAR-Based Global Localization of Mobile Robots Using Multiscale/Resolution Maps. *IEEE Transactions on Instrumentation and Measurement, 70*, 1-15.

Meshram, A. a. (2021). Pesticide spraying robot for precision agriculture: A categorical literature review and future trends. *Journal of Field Robotics*. doi:{10.1002/rob.22043}

Miao, Y. a. (2021). Airborne LiDAR Assisted Obstacle Recognition and Intrusion Detection Towards Unmanned Aerial Vehicle: Architecture, Modeling and Evaluation. *IEEE Transactions on Intelligent Transportation Systems, 22*, 4531-4540.

Nie, F. a. (2020). LCPF: A Particle Filter Lidar SLAM System With Loop Detection and Correction. *IEEE Access, 8*, 20401-20412.

Pham, T. T.-J. (2014). Interacting Geometric Priors For Robust Multimodel Fitting. *IEEE Transactions on Image Processing, 23*, 4601-4610.

Ramezani, M., Tinchev, G., Iuganov, E., & Fallon, M. (2020). Online lidar-slam for legged robots with robust registration and deep-learned loop closure. *IEEE International Conference on Robotics and Automation (ICRA).* doi:10.1109/ICRA40945.2020.9196769.

Rovira-Más, F. a.-R.-C. (2021). Augmented Perception for Agricultural Robots Navigation. *IEEE Sensors Journal, 21*, 11712-11727.

Syaqur, W. A., Ali-Yeon, A. S., Abdullah, A. H., Kamarudin, K., Visvanathan, R., & Ismail, A. H. (2018). Mobile Robot Based Simultaneous Localization and Mapping in UniMAP's Unknown Environment. *International Conference on Computational Approach in Smart Systems Design and Applications (ICASSDA)*, (pp. 1-5). doi:10.1109/ICASSDA.2018.8477629

Taher, M., et al. (2012). Control and analysis of an omnidirectional autonomous robot based on software approach and multimedia database. *Journal of Control Engineering and Applied Informatics*, 14(2), 1-8.

Tamas, L., et al. (2010). Lidar and vision based people detection and tracking. *Journal of Control Engineering and Applied Informatics*, 12(2), 30-35.

Wang, H., Wang, C., & Xie, L. (2021). Lightweight 3-D Localization and Mapping for Solid-State LiDAR. *IEEE Robotics and Automation Letters, 6*, 1801-1807.

Warren, M. E. (2019). 2019 Symposium on VLSI Circuits. *Automotive LIDAR Technology*, (pp. C254-C255). doi:10.23919/VLSIC.2019.8777993

Winterhalter, W. a. (2018). Crop Row Detection on Tiny Plants With the Pattern Hough Transform. *IEEE Robotics and Automation Letters, 3*, 3394-3401.

Xiong, H. a. (2009). K-Means Clustering Versus Validation Measures: A Data-Distribution Perspective. *International Journal of Distributed Sensor Networks, 39*, 318-331.

Xu, W. a. (2021). FAST-LIO: A Fast, Robust LiDAR-Inertial Odometry Package by Tightly-Coupled Iterated Kalman Filter. *IEEE Robotics and Automation Letters, 6*, 3317-3324.

Xu, Y. a. (2019). Indoor INS/LiDAR-Based Robot Localization With Improved Robustness Using Cascaded FIR Filter. *IEEE Access, 7*, 34189-34197.

Yousif, K., Bab-Hadiashar, A., & Hoseinnezhad, R. (2015). An Overview to Visual Odometry and Visual SLAM: Applications to Mobile Robotics. *Intelligent Industrial Systems, 1*.

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*, 165-175.

Zhang, Y. a. (2020). An Automatic Background Filtering Method for Detection of Road Users in Heavy Traffics Using Roadside 3-D LiDAR Sensors With Noises. *IEEE Sensors Journal, 20*, 6596-6604.