Development of a robot with 3D perception for accurate row following in vineyard

Yubin Lan[1,2], Lijie Geng[1], Wenhua Li[1], Weixu Ran[1], Xiang Yin[1], Lili Yi[1]

(1. *College of Agricultural Engineering and Food Science, Shandong University of Technology, Shandong Provincial Engineering Technology Research Center for Agricultural Aviation Intelligent Equipment, Zibo 255022, Shandong, China*; 2. *Zibo Agricultural and Rural Service Center, Zibo 255022, Shandong, China*)

Abstract: In this paper, a localization and navigation guidance algorithm based on the tree row in orchards for agricultural robots was proposed. In this algorithm, the tree rows formed by the trunks of parallel planting grape trees were used as auxiliary information. Together with the relative position provided by IMU, odometer and 3D LiDAR, the proposed algorithm was used to calculate the position and orientation of the moving robot. Firstly, the coordinates of the robot were obtained by the vehicle positioning system, and then the orientation of the tree and the posture of the robot relative to the tree row were determined through the 3D LiDAR point cloud, to ensure that the robot can parallel walk along the tree row and maintain a specific distance. Finally, the odometer, IMU data and LiDAR point cloud image were deeply fused to determine the accurate position and orientation to complete the path planning in the orchards. Experimental results show that the algorithm can effectively improve the positioning accuracy of the robot walking in the orchard and ensure that the robot can walk along the tree row without hitting the trees.

Keywords: Autonomous robot, tree row navigation, 3D LiDAR, Kalman filter, estimation, detection
DOI: 10.33440/j.ijpaa.20210402.177

Citation: Lan Y B, Geng L J, Li W H, Ran W X, Yin X, Yi L L. Development of a robot with 3D perception for accurate row following in vineyard. Int J Precis Agric Aviat, 2021; 4(2): 14–21.

1 Introduction

Orchards require a great deal of maintenance throughout the year, pruning, bloom thinning, spraying for insects and disease, and mowing the grass between the trees[1]. These activities make up a significant portion of operating expenses and improvements in efficiency can directly improve a vineyards productivity. More and more professional farmers recognize the potential of automation to reduce chemical exposure to their employees during spraying and help reduce the logistical difficulties of finding sufficient, skilled, seasonal labor.

In recent years, autonomous agriculture equipment has become more and more feasible with the development of navigation technologies including Micro-Electro-Mechanical System (MEMS)[2], Global Navigation Satellite System (GNSS)[3], Artificial Intelligence (AI) and so on. The GNSS receiver with multi-system, multi-band and Real-time Kinematic (RTK)[4] can improve the positioning accuracy to be several centimeters in the open field. Except that, the cost of the GNSS-RTK system is lower and lower with the development of network RTK. Consequently, many researchers put their attention on autonomous agriculture vehicles, especially on precision localization and automatic guidance.

However, the limitation of GNSS-RTK is also introduced huge positioning errors in Non-Line-of-Sight (NLOS)[5] environments since of the refraction and diffraction during the satellite signal propagation, such as in the orchards and during cloudy days[6]. Spatial information collected by Inertial Measurement Unit (IMU), 3D LiDAR[7] and RGB-D[8] camera is required to localize the agricultural vehicles as accurate as possible.

On most scenarios, agriculture robots are required to move in straight line and constant turning trajectories to cover all the target field. From the kinematics point of view, it can be decomposed as constant velocity (CV) and constant turning (CT) movement in the orchards. For constant velocity movement, autonomous row following has become a popular research area especially in agriculture application because of the standardized planting model[9-18]. The task consists of detecting a pathway for an agriculture vehicle to follow, using environmental sensors including 3D LiDAR, IMU and camera. For constant turning, it is still a research problem with difficulty.

Here, we address the problem of row following guidance for autonomous vehicles in orchards. A monocular camera is adopted to detect the crops planted in straight rows using Hough transform proposed by Astrand and Baerveldt in[19]. Satow also used Hough transform to detect the crop rows from 3D point cloud perceived by a laser scanner through a rotating mirror[20]. Biber also proposed a row following based navigation system using data collected from GNSS-RTK, an inertial sensor, a LiDAR and odometry[21]. Still other research results have been proposed in these references[22–27].

In these proposed algorithms, only the CV moving model is considered while the CT moving model is missing. But in actual applications, the CT moving model is as important as the CV

Received date: 2021-09-15 **Accepted date:** 2021-10-19

Biographies: Yubin Lan, PhD, Distinguished Professor, Director, research interests: precision agricultural aviation application,. Mailing Address: College of Electronic Engineering, South China Agricultural University. Email: ylan@scau.edu.cn; **Lijie Geng**, postgraduate, research interests: agricultural automation, Email: 1779333079@qq.com; **WenHua Li**, postgraduate, research interests: agricultural automation, Email: L17864306386@163.com; **WeiXu Ran**, postgraduate, research interests: agricultural automation, Email: 625973407@qq.com; **Xiang Yin**, PhD, research interests: agricultural automation and autonomy, Email: 666513@163.com.

***Corresponding author: Lili Yi, PhD**, research interests: agricultural automation and autonomy. Room 6-915, School of Agricultural Engineering and Food Science, Shandong University of Technology, Zibo 255000, China. Tel: +86-18369916558, Email: yili0001@sdut.edu.cn.

moving model. To overcome this weakness, we proposed a navigation algorithm considering both CV and CT moving models as a whole.

The objective of this research is to develop an autonomous navigation controller with vehicle positioning system consisting of multi sensors. The controller can provide global navigation path and local navigation path with multi-sensors. The LiDAR point cloud is registered by vehicle motion measured by wheel encoders and IMU. Then the point cloud is fitted into two parallel straight lines based on Least Square (LS) regression algorithm. The straight lines are integrated into an Extended Kalman Filter (EKF)[28] to reduce the line fitting noise. To follow the tree rows with a constant offset to one side of the trees, we also detect the closest tree trunk to the vehicle each side. After that, the controller also handles the situation of the end of tree row by hypothesis test algorithm to verify the vehicle is moving along the tree row or moving at the end of the tree row. While the vehicle is moving along the tree row, the CV moving model is adopted. While the vehicle is close to the end of row, the CV moving model is adopted. The simulation and experimental results are demonstrated that the proposed algorithm improved the navigation accuracy to 2-5 cm during the whole path with both straight-line and turning.

The rest of the paper is organized as follows. Section 2 describes the moving modes employed in this paper to guide the vehicle to move in the vineyard. Section 3 gives a detailed description of the proposed navigation controller and algorithms used in this paper. Section 4 describes all the experiments implemented in the vineyard and evaluated the experiments result with Stanley algorithm. Section 5 draws the conclusion of this paper.

2 Problem statement

From the robot point of view, the problem addressed in this paper is to detect the tree row and to guide the robot to go straight with constant velocity mode in between the tree row with a certain range to each side of the tree row. Except that, the beginning and the end of the tree row are also required to be detected by the 3D point cloud to guide the robot to change from one tree row to another with constant turning model as shown in Figure 1 and Figure 2, respectively.

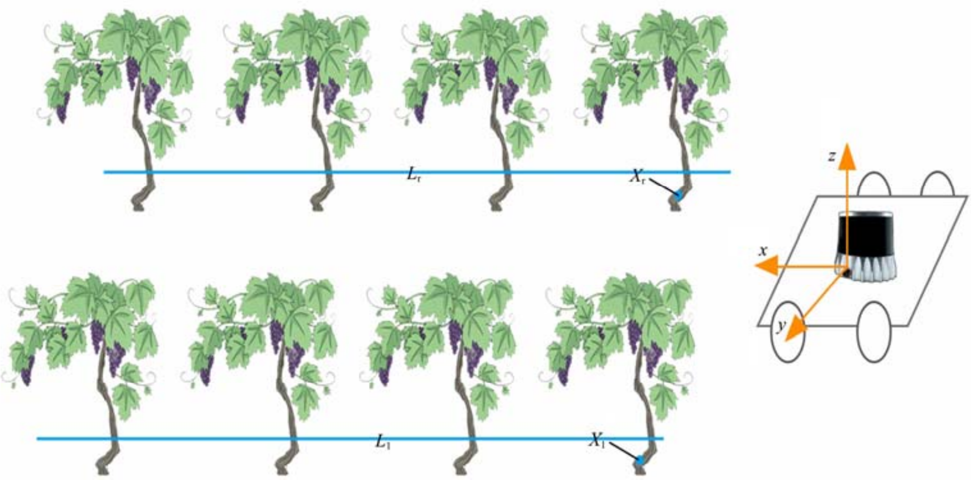


Figure 1 Problem statement. L_l and L_r are two parallel straight lines representing the tree rows, X_l and X_r are two points representing the centers of the closest tree trunks on both sides of the vehicle. Our task consists in detecting L_l, L_r, X_l, and X_r from the LiDAR point cloud in coordinate $\{x, y, z\}$

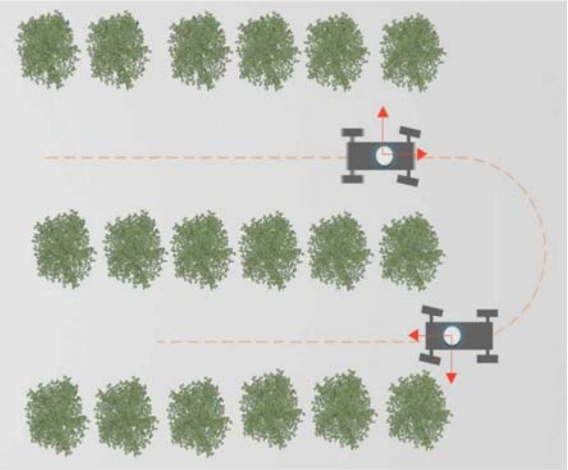


Figure 2 *Problem* statement. L_l and L_r are two parallel straight lines representing the tree rows, X_l and X_r are two points representing the centers of the closest tree trunks on both sides of the vehicle. Our task consists in detecting L_l, L_r, X_l, and X_r from the LiDAR point cloud in coordinate $\{x, y, z\}$

2.1 Dynamic models

Here we assume that the autonomous vehicle is moving with a Constant velocity model and coordinated turning are as shown in Figure 2. The state of the vehicle at time step k, $k \in Z^+$, is denoted as $x_k = [x_k \ y_k \ \dot{x}_k \ \dot{y}_k]^T$,

where, (x_k, y_k) is the instantaneous position and ($\dot{x}_k$, $\dot{y}_k$) is the instantaneous speed, respectively.

Under CV model, the following equations can be used to describe its evolution over time:

$$x_k = x_{k-1} + \dot{x}_{k-1}T \tag{1}$$

where, x_k is the position of a moving target, $x^{\cdot}$ is the velocity of the target and T is the time interval. Considering a target moving in a $2-D$ plane, the state vector can be organized as $\mathbf{x}_k = [x_k \ y_k \ \dot{x}_k \ \dot{y}_k]^T$, and the state-space evolution can be rewritten as:

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} \tag{2}$$

where the transformation matrix **F** is:

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

When the target is moving in a urban environment, the state-space evolution equation (2) is organized as:

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{\mathbf{k-1}} + \mathbf{v}_\mathbf{k} \tag{4}$$

where, $\mathbf{v}_\mathbf{k}$ is a Gaussian distributed process noise.

Under CT model, the evolution equations of the autonomous vehicle's velocities are:

$$\dot{x}_k = s_k \cos\phi_k \tag{5}$$

$$\dot{y}_k = s_k \sin\varphi_k \$ \tag{6}$$

where, $\dot{x}_k$ and $\dot{y}_k$ are the velocity components corresponding to x and y directions, $s_k = \sqrt{\dot{x}_k^2 + \dot{y}_k^2}$ is the target's speed and ϕ_k is the heading angle.

The turning rate equals to the first derivation of the heading angle:

$$\omega_k = \dot{\phi}_k \tag{7}$$

Considering constant turning model, the turning rate is fixed:

$$\omega_k = \omega_{k-1} \tag{8}$$

The equations describing the evolution the x position and position y with constant ω_k are:

$$\begin{aligned} dx_k &= x_{k-1} + \int_{t_{k-1}}^{t_k} \dot{x}_{\tau|\, t_{k-1}} d_\tau \\ &= x_{k-1} + s_k \int_0^T \cos(\phi_{k-1} + \omega_{k-1}\tau) d_\tau \\ &= x_{k-1} + \dot{x}_{k-1} \frac{\sin(\omega_{k-1}T)}{\omega_{k-1}} - \dot{y}_{k-1} \frac{1-\cos(\omega_{k-1}T)}{\omega_{k-1}} \end{aligned} \tag{9}$$

$$\begin{aligned} y_k &= y_{k-1} + \int_{t_{k-1}}^{t_k} \dot{y}_{\tau|\, t_{k-1}} d_\tau \\ &= y_{k-1} + s_k \int_0^T \sin(\phi_{k-1} + \omega_{k-1}\tau) d_\tau \\ &= y_{k-1} + \dot{x}_{k-1} \frac{1-\cos(\omega_{k-1}T)}{\omega_{k-1}} + \dot{y}_{k-1} \frac{1-\cos(\omega_{k-1}T)}{\omega_{k-1}} \end{aligned} \tag{10}$$

where, (x_k, y_k) is the position of the target; T is the time interval between the step $k-1$ and the step k.

Similarly, as these position evolution equations, the speed evolution equations are:

$$\begin{aligned} \dot{x}_k &= s_{k-1} \cos(\phi_{k-1} + \omega_{k-1}T) \\ &= \dot{x}_{k-1} \cos(\omega_{k-1}T) - \dot{y}_{k-1} \sin(\omega_{k-1}T) \end{aligned} \tag{11}$$

$$\begin{aligned} \dot{y}_k &= s_{k-1} \sin(\phi_{k-1} + \omega_{k-1}T) \\ &= \dot{x}_{k-1} \sin(\omega_{k-1}T) + \dot{y}_{k-1} \cos(\omega_{k-1}T) \end{aligned} \tag{12}$$

Defining the state vector $\mathbf{x}_k = [x_k \ \ y_k \ \ \dot{x}_k \ \ \dot{y}_k]^T$ introducing the process noise vector $\mathbf{v}$, the state transition equation of the dynamic system is:

$$\mathbf{x}_k = \mathbf{f}_{k-1}(\mathbf{x}_{k-1}) + \mathbf{v} \tag{13}$$

where, $\mathbf{f}_k$ is the nonlinear function:

$$\mathbf{f}_{k-1}(\mathbf{x}_{k-1}) = \begin{bmatrix} x_{k-1} + \dot{x}_{k-1} \dfrac{\sin(\omega_{k-1}T)}{\omega_{k-1}} - \dot{y}_{k-1} \dfrac{1-\cos(\omega_{k-1}T)}{\omega_{k-1}} \\ y_{k-1} + \dot{x}_{k-1} \dfrac{1-\cos(\omega_{k-1}T)}{\omega_{k-1}} + \dot{y}_{k-1} \dfrac{1-\cos(\omega_{k-1}T)}{\omega_{k-1}} \\ \dot{x}_{k-1} \cos(\omega_{k-1}T) - \dot{y}_{k-1} \sin(\omega_{k-1}T) \\ \dot{x}_{k-1} \sin(\omega_{k-1}T) + \dot{y}_{k-1} \cos(\omega_{k-1}T) \\ \omega_{k-1} \end{bmatrix} \tag{14}$$

2.2 Stanley method

To evaluate the performance of the autonomous vehicle, the cross-track error $e(t)$ and the angle of those wheels with respect to the nearest segment of trajectory are introduced.

The Stanley method is a nonlinear feedback function of the cross-track error $e(t)$, measured from the center of the front axle to the nearest path point (p_x, p_y). Co-locating the point of control with the steered front wheels allows for an intuitive control law, where the first term simply keeps the wheels aligned with the given path by setting the steering angle δ equal to the heading error.

The heading error is calculated by Equation(15).

$$\theta_e = \theta - \theta_p \tag{15}$$

where, e is the lateral error; θ is the heading of the vehicle and θ_p is the heading of the path at (p_x, p_y). When e is zero, the direction of the front wheel deflection angle is consistent with the direction of the tangent of the given path. When e is non-zero, the second term adjusts δ such that the intended the path tangent from (p_x, p_y) at $kv(t)$ units from the front axle. Figure 3 illustrates the geometric relationship of the control parameters. The resulting steering control law is calculated by Equation(16),

$$\begin{aligned} \delta(t) &= \theta_e(t) + \delta_e(t) \\ &= \theta_e(t) + \arctan(\frac{e(t)}{d(t)}) \\ &= \theta_e(t) + \arctan(\frac{ke(t)}{v(t)}) \end{aligned} \tag{16}$$

where, $d(t)$ is the distance between the expected trajectory and the front axle; $v(t)$ is the vehicle speed; k is a gain parameter. It is clear that the desired effect is achieved with this control law: As e increases, the wheels are steered further towards the path.

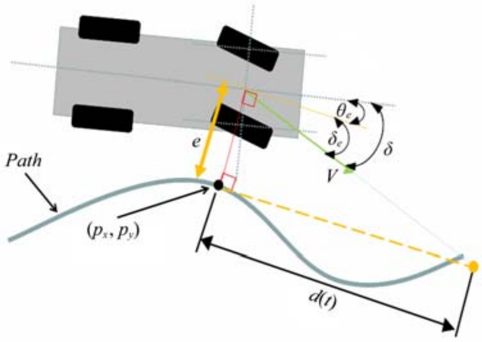


Figure 3 Stanley method geometry

Using a linear bicycle model with infinite tire stiffness and tight steering limitations results in the following effect of the control law,

$$\dot{e}(t) = -v(t) \sin\delta_e(t) \tag{17}$$

where, $\sin\delta_e(t)$ can be seen from the geometric relationship:

$$\sin\delta_e(t) = \frac{e(t)}{\sqrt{d(t)^2 + (e(t))^2}} = \frac{ke(t)}{\sqrt{v(t)^2 + (ke(t))^2}} \tag{18}$$

Therefore, Equation (17) can be simplified to Equation (19).

$$\dot{e}(t) = \frac{-v(t)ke(t)}{\sqrt{v(t)^2 + (ke(t))^2}} = \frac{-ke(t)}{\sqrt{1 + (\frac{ke(t)}{v(t)})^2}} \tag{19}$$

When the lateral tracking error $e(t)$ is very small, that is, $(\frac{ke(t)}{v(t)})^2$ converges to 0, the Equation (19) can be further simplified to Equation (20).

$$\dot{e}(t) \approx -ke(t) \tag{20}$$

Under the integral equation, the lateral tracking error is calculated by Equation(21).

$$\dot{e}(t) \approx e(0) \times \exp^{-kt} \tag{21}$$

Thus, the error converges exponentially to $e(t)$=0. The parameter k determines the rate of convergence. As cross-track error increases, the effect of the arctan function is to turn the front wheels to point straight toward the trajectory, yielding convergence limited only by the speed of the vehicle. For any value of $e(t)$, the differential equation converges monotonically to zero.

3 Materials and methods

3.1 System overview

A robot with multi-sensor positioning system is employed to verify the algorithm proposed in this paper. The multi-sensor positioning system consists of a 64 channels 3D LiDAR, which has a vertical field of view of 45 degrees with 2.8 degrees resolution and 10 Hz frame rate, an IMUS and an odometer. Encoder and IMU data are used to verify the 3D point cloud information. The detailed information of three-dimensional radar, odometer and other sensors are shown in Table 1.

Table 1 Detailed information of three-dimensional radar, odometer and other sensors

	Parameter	Value
Lidar	Precision	±0.7-5 cm
	Frequency	10 Hz
	Detection distance	120 m
	Horizontal resolution	1024
IMU	Roll and pitch accuracy	0.25°
	Heading accuracy	1°
	Frequency	100 Hz
Encoder	Resolution	1000P/R
	Frequency	100 Hz

The experiment was conducted in a standardized vineyard in Zibo, Shandong Province. Weeds and vines in the vineyard are regularly pruned, so Robot Perception and navigation are not affected. In order to facilitate the robot to have good traffic capacity in special environment such as muddy environment, the robot was designed with crawler chassis and driven by stepping motor.

Considering that the grapevine stalk point cloud above 0.5 m from the access road is relatively clean and less affected by weeds. The lidar was installed in the middle of the front end of the robot chassis, with a height of about 1.0 m from the ground, to adopt point cloud data more than 0.5 m from the ground.

The autonomous navigation system, worked as the main processor part of the robot, is used to process the 3D LiDAR cloud point information, to implement the navigation algorithms, and to manage the navigation information. The architecture of the navigation system is shown in Figure 4. The navigation controller is composed of a main processor NVIDIA Jetson AGX Xavier and an auxiliary processor embedded STM32F429 micro-controller.

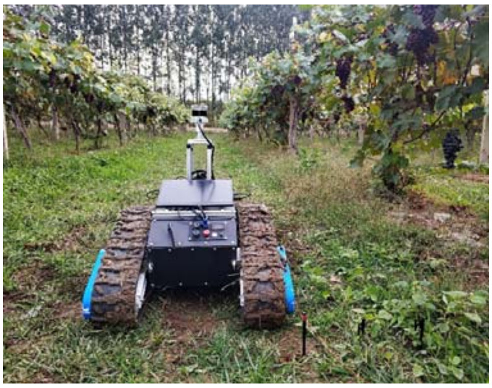


Figure 4 The robot collects three-dimensional point clouds in the vineyard

The main processor AGX is used to run the Robot Operating System and 3D point cloud processing algorithms. The main processor is also used to connect to the LCD screen and the keyboard. The auxiliary processor is used to control the motors, to read the encoder data. Serial port is used to communicate in between the main processor and the auxiliary processor.

Multiple sensors are synchronized based on the time of the 3D LiDAR. When the controller obtains the LiDAR data through the network port, the odometer data of other sensors under the point cloud time is obtained by linear interpolation. The point cloud is registered by odometer data, and then the point cloud registration is optimized by Normal-Distribution Transformation (NDT). Then two lines are fitted by Random Sample Consensus (RANSAC) method based on the least square method, and the lines are filtered by EKF algorithm with the odometer calculated before as the input value. The software architecture and the data process are shown in Figure 4 and Figure 5, respectively.

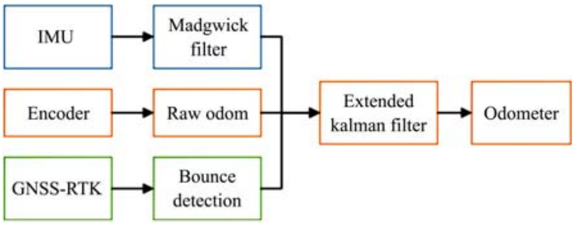


Figure 5 Multi-sensors data fusion

3.2 Multi-sensor data fusion

To use the information collected by different sensors, the multi-sensor data fusion is implemented at the very beginning of the process. Then the point cloud information is used to get the tree row. Finally, the robot trajectory is determined. The data processing algorithm is shown in Figure 6.

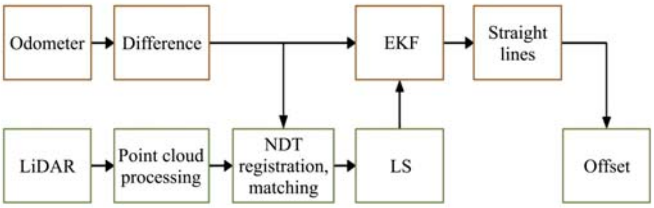


Figure 6 Data processing algorithms

The data of odometer is provided by IMU, encoder and RTK equipment. The output data of IMU has been filtered, and the data is relatively stable and reliable. To ensure the stability of IMU data on the bumpy ground in the field, the Marwick algorithm is used to filter again. The encoder data is the measured value of the driving wheels on both sides of the differential drive. Based on the differential motion model, the original odometer data is calculated with the encoder data.

Considering that the radar works at 10 Hz, IMU and encoder work at 100 Hz. Taking the radar data receiving time as the standard, the IMU and encoder data at the same time through the time-synchronizer function to solve the problem of data loss caused by multi-sensor fusion.

The original odometer will be affected by wheel slip and friction, and the positioning accuracy of RTK will be reduced due to the influence of environment, while the IMU data will have a long time drift accumulation. Therefore, the three-odometer data with better stability and more accurate data can be obtained through EKF fusion. The odometer track tested in the vineyard after fusion is as shown in Figure 7.

The output data of LiDAR includes ground point cloud, environmental noise, etc, in which ground point cloud is not only useless, but also affects the straight-line fitting effect. Considering the height of ground weeds and tree trunk, the point cloud above 0.5 m on the ground is taken as the input point cloud of straight-line fitting. Firstly, the point cloud is passed through the current odometer difference Δk for conversion as follows:

$$P_k=RP_k+T \tag{22}$$

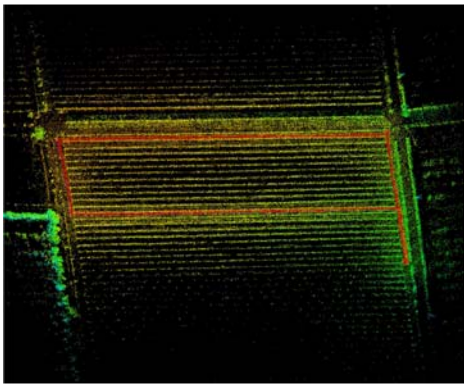


Figure 7 Point cloud of the vineyard at Zibo City of Shandong Province in China

3.3 Tree row detection algorithm

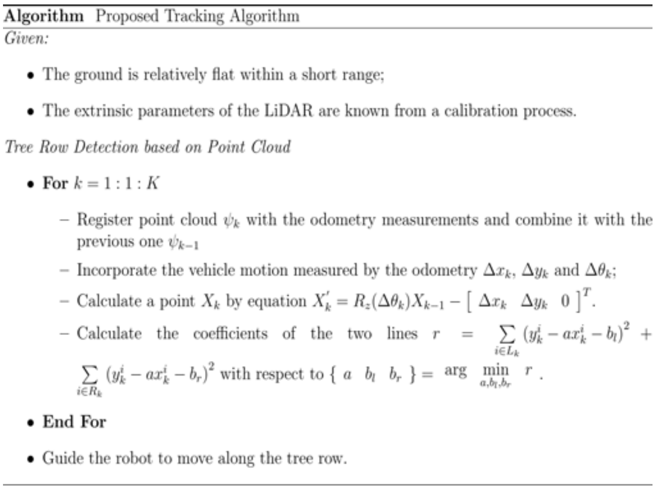
To depict the actual environments of the vineyard with grape trees in the tree row, an extracted deal picture is shown in Figure 1. As shown in Figure 1, L_1 and L_r stand for tow horizontal, parallel lines in the vehicle coordinate system on the left and right sides of the vehicle, respectively. X_1 and X_r stand for centers of the tree trunks closest to the vehicle on left and right side, respectively.

The information about the 3D point cloud is processed by a RANSAC algorithm to yield the supporting lines of the tree rows. L_l and L_r are represented as,

$$\begin{aligned} L_\mathrm{l}:\quad & y = ax + b_l \\ L_\mathrm{r}:\quad & y = ax + b_r \end{aligned} \tag{23}$$

where, a, b_l and b_r are the line coefficients. To determine these coefficients, a Least Square Line Regression (LSLR) algorithm is used to process these cloud point data.

The algorithm is summarized as follows.



```
Algorithm  Proposed Tracking Algorithm
Given:
  • The ground is relatively flat within a short range;
  • The extrinsic parameters of the LiDAR are known from a calibration process.
Tree Row Detection based on Point Cloud
  • For $k = 1 : 1 : K$
      – Register point cloud $\psi_k$ with the odometry measurements and combine it with the previous one $\psi_{k-1}$
      – Incorporate the vehicle motion measured by the odometry $\Delta x_k$, $\Delta y_k$ and $\Delta\theta_k$;
      – Calculate a point $X_k$ by equation $X'_k = R_z(\Delta\theta_k)X_{k-1} - [\ \Delta x_k \ \ \Delta y_k \ \ 0\ ]^T$.
      – Calculate the coefficients of the two lines $r = \sum_{i\in L_k}(y_k^i - ax_k^i - b_l)^2 + \sum_{i\in R_k}(y_k^i - ax_k^i - b_r)^2$ with respect to $\{\ a \ \ b_l \ \ b_r\ \} = \arg\min_{a,b_l,b_r} r$ .
  • End For
  • Guide the robot to move along the tree row.
```

To detect when to close to the end of tree row, hypothesis tests are incorporated into the navigation algorithm. At the very beginning of each tree row, two hypotheses are proposed as follows to test the number of the cloud point to identify whether the robot is close to the end of the tree row.

$$\begin{aligned} H_0:\quad & |N_k - N_{k-1}| \le N_{CONST} \\ H_1:\quad & |N_k - N_{k-1}| > N_{CONST} \end{aligned} \tag{24}$$

where, N_k is the number of cloud point in a certain range with 3 m wide around the robot. If hypothesis H_0 is correct, it indicates that the robot is running in the tree row but not close to the end, otherwise, the robot is close to the end of the tree row. When the robot is close to the tree row, another two hypothesis tests are proposed as follows:

$$\begin{aligned} H_0:\ & N_k > 0 \\ H_1:\ & N_k \to 0 \end{aligned} \tag{25}$$

the number of cloud point in a certain range with 0.1 m wide around the robot is used as the resolution. If hypothesis H_0 is correct, it indicates that the robot is running close to the end, otherwise, the robot is to the end of the tree row.

4 Results and discussion

To verify the 3D perception algorithm proposed in this paper, experiments are implemented in the vineyard. 3D point cloud data are collected by a 64-channel Ouster 3D LiDAR. Details of the experiments are described in this section. To test the tree row detection algorithm, the robot was controlled to move along the centerline of tree row with a speed of 1 m/s. The average positioning error is within 5 cm in a 260 cm wide row.

Figure 8 shows an example of the detected tree row while the data is provided by 3D LiDAR sensor. Off-line tests were conducted with data collected over 2 hours in one commercial vineyard in China.

The 3D LiDAR is mounted on a remote control robot with constant velocity 1 m/s. An example of the results obtained is shown in Figure 8. Three different parts of the tree row is selected to verify the results. Each of the straight-line fitting is run more than 1000 times. The root mean square error (RMSE) is summarized as shown in Table 2.

From Table 2, it can be seen that the calculated experimental mean of the parameters of each piece of the straight under different position of the tree row is very close to the true position of the tree row. The RMSE of the distance between the true value and the calculated value to the central line of the tree row is not greater than 0.2 cm which is negligible for the moving robot.

Taking the vineyard scene as an example, the walking trajectory is generated by using the 3D LiDAR point cloud data and attitude sensor data in the process of robot navigation. The experimental navigation effect is shown in Figure 9a.

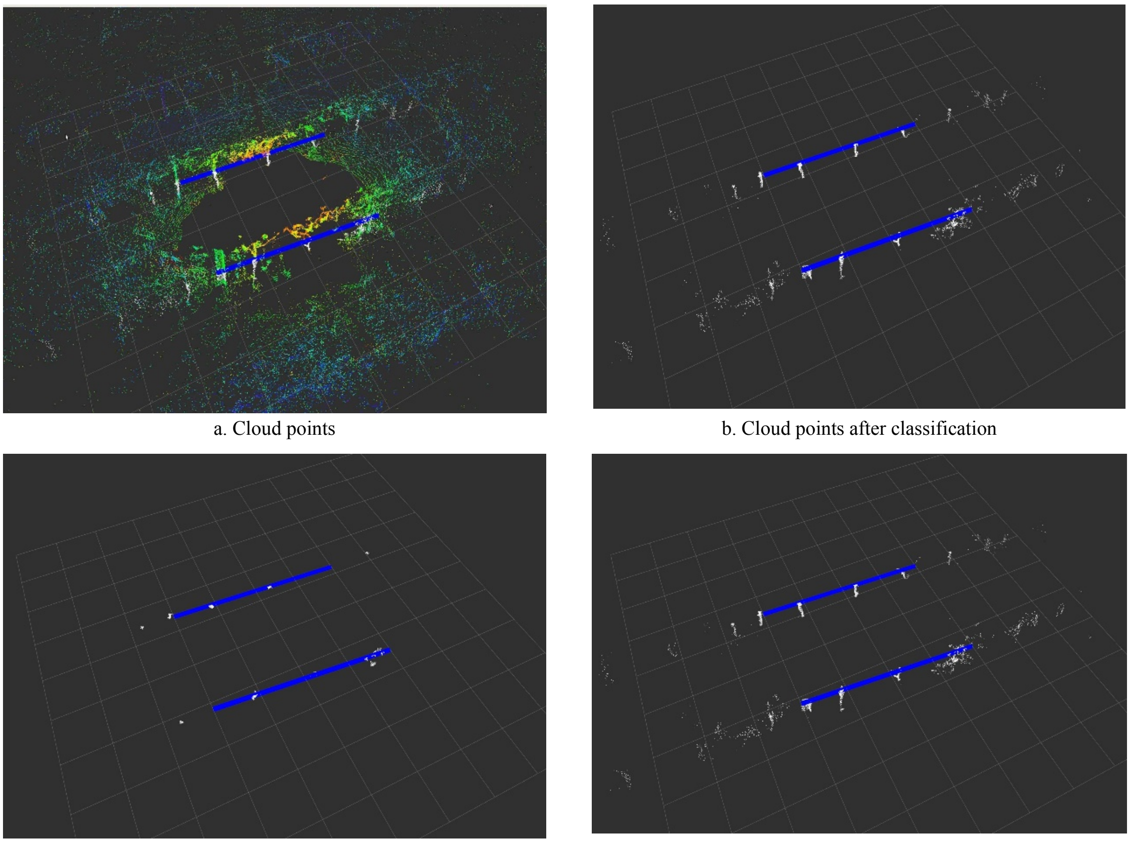
Table 3 shows the results of lateral error and heading error during the test. It can be seen from Table 3 that in the navigation test, the variance of lateral error is 0.00049 m, the standard deviation of lateral error is 0.022 m, and the absolute average value of lateral error is 0.0186 m. In the navigation test, the variance of heading error is 5.81°, the standard deviation of heading error is 2.41° and the absolute average value of heading error is 1.54°.

The lateral error of the navigation track in the test is shown in Figure 9b, and the heading error is shown in Figure 9c.

The small amplitude oscillation of the test track in Figure 9b is due to the large vibration of the robot chassis affected by the terrain, which makes the lidar located on the chassis vibrate slightly, resulting in measurement error. The large amplitude oscillation in Figure 9c is due to the small differential coefficient and insufficient damping during PID control navigation, resulting in no rapid stabilization in the oscillation. At the same time, due to the absence of piecewise PID control and the navigation control is not fine enough. In the follow-up research, the method of installing PTZ and segmented PID control will be used to solve the above two vibration problems.

The test results in Figure 9 and Table 2 show that the localization and navigation guidance algorithm based on the tree row in orchards proposed can make the robot trajectory evenly distributed along the midline, and tend to be stable. The algorithm basically meets the requirements of inter row autonomous operation.

Figure 10a shows the original point cloud data. The data 0.5-1.3 m high from the ground in the original point cloud data was intercepted as the turning detection data, as shown in Figure 10b. During automatic route tracking and navigation, the unmanned vehicle continuously detects the number of point clouds *n* on the left and right sides in the range of 3 m to 5 m ahead. When $n<30$, the unmanned vehicle is considered to have detected the end and began to plan the turning route, and the left and right turning actions are carried out alternately. In addition, considering that the row spacing of vines in the experimental site is about 3 m, the fixed turning radius was set as 1.5 m. The point cloud effect of unmanned vehicle turning test is shown in Figure 10c.



a. Cloud points

b. Cloud points after classification

c. Left side of the tree row

d. Right side of the tree row

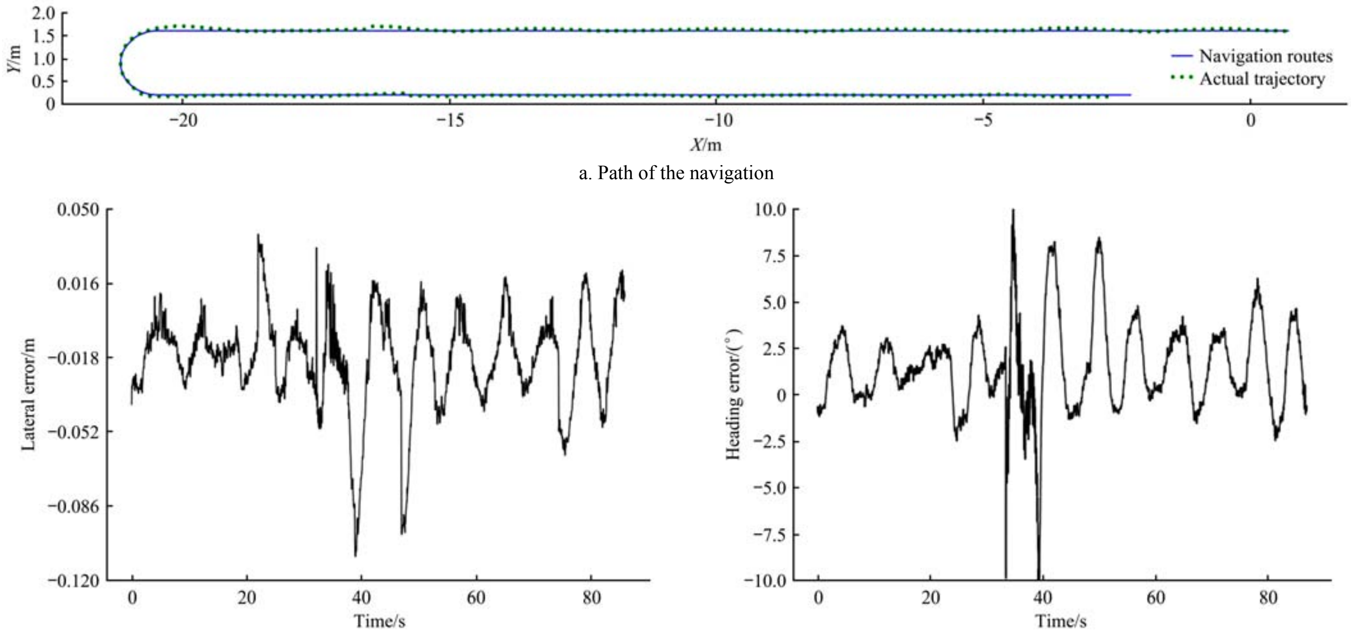
Figure 8 Cloud points in the vineyard and the tree row extracted from the cloud point

Table 2 Root mean square error of each piece for tree row in the vineyard

	Random trajectory, cellular network					
Algorithms	Right Slope	Right Intercept	Right Distance	Left Slope	Left Intercept	Left Distance
Average	At the starting point of the tree row					
	–0.008780	2.248745	2.243300	–0.037167	–2.031682	2.032948
RMSE	0.04345	0.097	0.0977	0.0242	0.0613	0.0604
Average	At the middle of the tree row					
	–0.036904	1.350937	1.350937	–0.048399	–1.707660	1.705493
RMSE	0.039	0.091	0.091	0.063	0.1076	0.107
Average	At the end of the tree row					
	–0.009662	1.339421	1.338286	–0.003152	–1.540490	1.538952
RMSE	0.0404	0.1052	0.1054	0.044	0.1407	0.1403

Table 3 Test error statistics

	Error statistics	
Parameter	Lateral error	Heading error
Variance	0.00049 m	5.81°
Standard deviation	0.022 m	2.41°
Absolute values of means	0.0186 m	1.54°

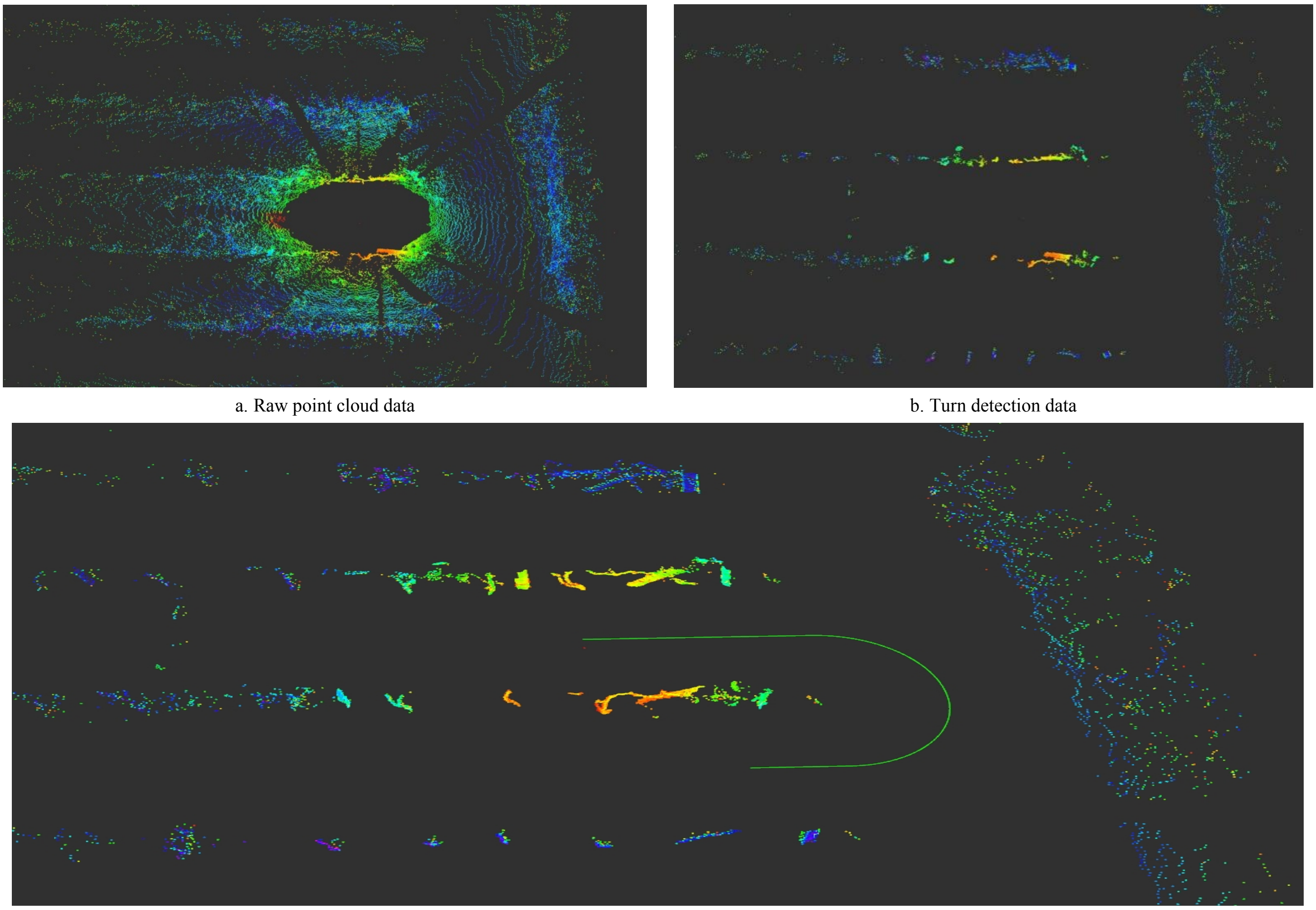


a. Path of the navigation

b. Lateral error

c. Heading error

Figure 9 Robot walking trajectory and cross tracking error



a. Raw point cloud data

b. Turn detection data

c. Planned turning route

Figure 10 Robot turning test

5 Conclusions

This paper has proposed a novel navigation algorithm based on the tree row in the vineyard. The LS has been used to estimate the pseudo-measured direction of a moving robot from 3D point cloud collected by the 64-channel LiDAR. As the Kalman filter is also implemented to track the estimation of the straight line's slope. The proposed algorithm has outperformed some state-of-the-art algorithms in terms of accuracy, particularly in situation where the NLOS condition is severe. We have also verified the effectiveness of the proposed tracking algorithm by simulations. Moreover, it has also been shown by simulations that the proposed algorithm is computationally efficient. Future work will be on the more challenging area of extending the current algorithm for other conditions.

[References]

[1] Moorehead S, Wellington C, Gilmore B, Vallespi C. Automating Orchards: A System of Autonomous Tractors for Orchard Maintenance. In: Proceedings of International Scientific Conference. 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2012.

[2] Abe Y, Kamiya K, Osaki T, Kawano R, Miki N, Takeuchi S. Proceedings of the IEEE International Conference on Micro Electro Mechanical Systems (MEMS). In: Proceedings of International Scientific Conference "IEEE International Conference on Micro Electro Mechanical Systems", 2007.

[3] Wendeberg J, Schindelhauer C. Polynomial-time approximation algorithms for anchor-free TDoA localization. Theoretical Computer Science, 2014, 553: 27–36. doi: 10.1016/j.tcs.2014.04.007.

[4] Zhao S, Chen Y M, Jay A F. High-Precision Vehicle Navigation in Urban Environments using a Low-cost Single-frequency RTK GPS Receiver. In: Proceedings of International Scientific Conference "2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)", 2013. doi: 10.1109/TITS.2016.2529000.

[5] Jasurbek K, Park Y, Aamir S M. Survey of NLOS identification and error mitigation problems in UWB-based positioning algorithms for dense environments. annals of telecommunications, 2010, 65(5): 301–311. doi: 10.1007/s12243-009-0124-z.

[6] Yi L L, Sirajudeen G R, Lin Z P, See C M. Target tracking in mixed LOS/NLOS environments based on individual TOA measurement detection. IEEE Transactions on Wireless Communications, 2014, 13(1): 99–111. doi: 10.1109/SAM.2010.5606720.

[7] Andrew B D, Gregory P A. Advances in animal ecology from 3D-LiDAR ecosystem mapping. Trends in Ecology & Evolution, 2014, 29(12): 681–691. doi: 10.1016/j.tree.2014.10.005.

[8] Hu J S, Wang J J, Daniel M H. Design of Sensing System and Anticipative Behavior for Human Following of Mobile Robots. in 2014 IEEE Transactions on Industrial Electronics, 2014, 61(4): 1916–1927. doi: 10.1109/TIE.2013.2262758.

[9] Zhang J, Andrew D C, Silvio M M, Marcel B, Sanjiv S. 3D perception for accurate row following: Methodology and results. In: Proceedings of International Scientific Conference "2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2013: 5306-5313. doi: 10.1109/IROS.2013.6697124.

[10] Kong H, Jean Y A, Jean P. General Road Detection From a Single Image. in IEEE Transactions on Image Processing, 2010, 19(8): 2211–2220. doi: 10.1109/TIP.2010.2045715.

[11] Zhang X Y, Liu P B, Zhang C X. An Integration Method of Inertial Navigation System and Three-Beam Lidar for the Precision Landing. Mathematical Problems in Engineering, 2016: 1–13. doi: 10.1155/2016/4892376.

[12] Paul D G. Principles of GNSS, inertial, and multisensor integrated navigation systems. in IEEE Aerospace and Electronic Systems Magazine 2015, 30(2): 26–27. doi: 10.1109/MAES.2014.14110.

[13] Han H Z, Wang J, Wang J J, Tan X L. Performance Analysis on Carrier Phase-Based Tightly-Coupled GPS/BDS/INS Integration in GNSS Degraded and Denied Environments. Sensors, 2015, 15(4): 8685–8711. doi: 10.3390/s150408685.

[14] Liu L, Tao M, Niu R X, Wang J, et al. RBF-Based Monocular Vision Navigation for Small Vehicles in Narrow Space below Maize Canopy. Applied Sciences, 2016; 6(6):182. doi: 10.3390/app6060182.

[15] Neda N, Rene J L, Cheng J H, Denis G. A New Technique for Integrating MEMS-Based Low-Cost IMU and GPS in Vehicular Navigation. Journal of Sensors, 2016, 2016(4): 1–16. doi: 10.1155/2016/5365983.

[16] Beau T, Lee D J, Kirt D L, James K A. Review of stereo vision algorithms and their suitability for resource-limited systems. Journal of Real-Time Image Processing, 2016, 11(1): 5–25. doi: 10.1007/s11554-012-0313-2.

[17] Jaemin B, Beom S S, Lee J H. Toward Accurate Road Detection in Challenging Environments Using 3D Point Clouds. ETRI Journal, 2015, 37(3): 606–616. doi: 10.4218/etrij.15.0113.1131.

[18] Naoum T, Dimitrios B, Dionysis B. AgROS: A Robot Operating System Based Emulation Tool for Agricultural Robotics. Agronomy, 2019, 9(7): 403. doi: 10.3390/agronomy9070403.

[19] Björn Å, Albert J B. A vision based row-following system for agricultural field machinery. Mechatronics, 2006, 15(2): 251–269. doi: 10.1016/j.mechatronics.2004.05.005

[20] Satow T, Matsuda K, Ming S B, Hironaka, K, Tan D. Development of laser crop row sensor for automatic guidance system of implements. In: Proceedings of International Scientific Conference Automation Technology for Off-road Equipment Conference, 2004. doi: 10.13031/2013.17826.

[21] Biber P, Ulrich W, Albert, A. Navigation system of the autonomous agricultural robot bonirob. In: Proceedings of International Scientific Conference 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2012.

[22] Pieter M B, Koen V B, Frits K V E, Joris I, Gook H K. Robot navigation in orchards with localization based on Particle filter and Kalman filter. Computers and Electronics in Agriculture, 2019, 157: 261–269. doi: 10.1016/j.compag.2018.12.046

[23] Zhang S, Guo C Y, Gao Z N, Adilet S, Chen J. Research on 2D Laser Automatic Navigation Control for Standardized Orchard. Applied Sciences, 2020, 10(8): 2763. doi: 10.3390/app10082763.

[24] Li Y W, Wang X J, Liu D X. 3D Autonomous Navigation Line Extraction for Field Roads Based on Binocular Vision. Journal of Sensors, 2019, 2019(8): 1–16. doi: 10.1155/2019/6832109.

[25] Bruna P P, Jose P M. Path errors in sugarcane transshipment trailers. Engenharia Agrícola, 2020, 40(2): 223–231. doi: 10.1590/1809-4430-Eng.Agric.v40n2p223-231/2020.

[26] Lyu H K, Park C H, Han D H, Kwak S, Choi B. Orchard Free Space and Center Line Estimation Using Naive Bayesian Classifier for Unmanned Ground Self-Driving Vehicle. Symmetry, 2018, 10(9): 1–14. doi: 10.3390/sym10090355.

[27] Hao Z, Xiong H L, Liu Y, Tan N D, Chen L. Trajectory Planning Algorithm of UAV Based on System Positioning Accuracy Constraints. Electronics, 2020, 9(2): 250. doi: 10.3390/electronics9020250.

[28] Sebastian T. Fox D. Probabilistic Robotics. Communications of the ACM, 2002, 45(3): 52–57. doi: 10.1145/504729.504754.