

Article

UAV-Borne LiDAR Crop Point Cloud Enhancement Using Grasshopper Optimization and Point Cloud Up-Sampling Network

Jian Chen ¹, Zichao Zhang ¹, Kai Zhang ¹, Shubo Wang ¹ and Yu Han ^{2,*}

¹ College of Engineering, China Agricultural University, Beijing 100083, China; jchen@cau.edu.cn (J.C.); zhangzc1@cau.edu.cn (Z.Z.); sy20193071234@cau.edu.cn (K.Z.); wangshubo@cau.edu.cn (S.W.)

² College of Water Resources & Civil Engineering, China Agricultural University, Beijing 100083, China

* Correspondence: yhan@cau.edu.cn

Received: 5 September 2020; Accepted: 28 September 2020; Published: 1 October 2020



Abstract: Because of low accuracy and density of crop point clouds obtained by the Unmanned Aerial Vehicle (UAV)-borne Light Detection and Ranging (LiDAR) scanning system of UAV, an integrated navigation and positioning optimization method based on the grasshopper optimization algorithm (GOA) and a point cloud density enhancement method were proposed. Firstly, a global positioning system (GPS)/inertial navigation system (INS) integrated navigation and positioning information fusion method based on a Kalman filter was constructed. Then, the GOA was employed to find the optimal solution by iterating the system noise variance matrix Q and measurement noise variance matrix R of Kalman filter. By feeding the optimal solution into the Kalman filter, the error variances of longitude were reduced to 0.00046 from 0.0091, and the error variances of latitude were reduced to 0.00034 from 0.0047. Based on the integrated navigation, an UAV-borne LiDAR scanning system was built for obtaining the crop point. During offline processing, the crop point cloud was filtered and transformed into WGS-84, the density clustering algorithm improved by the particle swarm optimization (PSO) algorithm was employed to the clustering segment. After the clustering segment, the pre-trained Point Cloud Up-Sampling Network (PU-net) was used for density enhancement of point cloud data and to carry out three-dimensional reconstruction. The features of the crop point cloud were kept under the processing of reconstruction model; meanwhile, the density of the crop point cloud was quadrupled.

Keywords: UAV-borne LiDAR scanning system; grasshopper optimization algorithm; GPS/INS integrated navigation; point cloud up-sampling network (PU-net); clustering segmentation; 3-dimensional reconstruction

1. Introduction

Light Detection and Ranging (LiDAR) is an active sensing technology that can quickly acquire spatial information of a target or environment [1,2]. Compared with traditional remote sensing methods, LiDAR has shown great advantages in accuracy and stability [3]. In agriculture, the LiDAR system is one of the most accurate methods to measure regional vegetation structural characteristics and biophysical parameters [4]. The Unmanned Aerial Vehicle (UAV)-borne LiDAR scanning system is one of the most common platforms for crop point cloud obtaining. The crop point clouds captured by UAV-borne LiDAR scanning system are sparse and unordered [5]. The navigation and positioning accuracy of the UAV-borne LiDAR scanning system influences the accuracy of the point cloud data greatly. Density is also an important indicator to measure the quality of point cloud data. Higher point cloud density represents richer information of the target or environment [6]. Ensuring the accuracy

and density characteristics of the UAV-borne LiDAR point cloud data while maintaining the cost is crucial for the continuous and efficient execution of LiDAR detection tasks.

For integrated navigation, the most widely used in the world is the global positioning system (GPS)/inertial navigation system (INS) integrated navigation positioning system [7]. The Kalman filter is the most common method of data fusion [8]. Gao et al. [9] developed an INS/GPS/LiDAR integrated navigation system through an extended Kalman filter (EKF) using a hybrid scan matching algorithm. Shabani et al. [10] explored the characteristics, advantages, and disadvantages of direct Kalman filtering and indirect Kalman filtering commonly used in GPS/INS integrated navigation, and a novel asynchronous direct Kalman filter was proposed. Zhang et al. [11] developed a low-cost global navigation satellite system (GNSS) and INS integrated navigation method and an adaptive Kalman filter for overcoming the measurement noise of GNSS, which was proposed based on a supervised machine learning model. Yu et al. [12] used a single-frequency real-time kinematic carrier phase (RTK) technology to deduce and obtain information including speed and orientation under GPS/BeiDou Navigation Satellite System (BDS) for polishing the inaccurate INS information. Norouz et al. [13] proposed an improved Kalman filter algorithm based on small probability distribution of the state variable between the measurement updates of the multi-rate GPS/INS-coupled navigation filter, which greatly reduced the amount of calculation, and the algorithm performed well by numerically simulating. All of the above integrated navigation systems suffered from the low accuracy of integrated navigation, except Yu et al. [12]. However, RTK-based high accuracy integrated navigation systems cannot be widely popularized due to the price gap. During the experiments of data fusion using the Kalman filter, the longitude and latitude errors were sharply affected by the parameter of the Kalman filter, especially system noise sequence Q and measurement noise sequence R , and the same situation also occurred in the realization of Ref. [10,11]. For promoting the accuracy of our integrated navigation-based scanning system, the meta-heuristic method was employed for finding the optimum parameters of integrated navigation in our UAV scanning system. For multi-solution problems, genetic algorithms (GA) [14], particle swarm optimization (PSO) [15], and ant colony optimization (ACO) [16] were the most popular, with swarm intelligence optimization proposed recently such as the bat algorithm (BA) [17] and grasshopper optimization algorithm (GOA) [18]. All of the multi-solution meta-heuristic methods have their own merits for different problems [19]. In order to reduce longitude and latitude errors of integrated navigation, the meta-heuristic methods were employed to find the optimum system noise sequence Q and measurement noise sequence R .

After obtaining the crop point cloud data by UAV scanning system, the raw point cloud data consisted of crop points and other points, and manual classification was extremely inefficient for different point cloud data. Therefore, the clustering segmentation method was employed to extract crop data from raw point cloud data. The density-based clustering method, density-based spatial clustering of applications with noise (DBSCAN), has attracted a lot of attention; besides the improvement of the algorithm itself, finding out the key parameters of clustering was also a new direction of optimization [20]. The same as integrated navigation, the meta-heuristic method was employed to optimize the clustering segmentation method. According to the experimental results of the density distribution characteristics of the plane point cloud under different conditions, Kedzierski et al. [21] found that the point cloud density could be preliminarily estimated based on distance and angle, but the actual point cloud density should take into account the influence of object shape and environment. Rupink et al. [22] proposed the formula and calculation steps for calculating the point cloud density. Huang et al. [23] proposed an edge-sensing point set resampling method, gradually approaching the edges and corners. However, the quality of the results largely depends on the accuracy of the normal of a given point and the fine-tuning of parameters. Based on Ref. [21,23], dense and accurate point clouds are very useful for crop point cloud recognition and 3D reconstruction. The high-cost multi-line LiDAR can be simulated by low-cost single-line LiDAR point clouds through the point cloud up-sampling method without destroying the features.

In view of the current problems and inspired by existing methods, in order to improve the quality of point cloud data, a combined navigation optimization algorithm based on GOA and a point cloud density enhancement algorithm based on Point Cloud Up-Sampling Network (PU-net) were proposed. The novel contribution of this article was as follows:

- (1) In order to obtain the optimal solution of the initial parameters of the Kalman filter quickly and accurately, GOA took the sum of the longitude and latitude error variances as the fitness function to find the optimal solution to iterate the system noise variance matrix Q and the measurement noise variance matrix R . Based on the Q and R matrices from GOA, the positioning error of the UAV-borne LiDAR scanning system was greatly reduced; meanwhile, the computing burden was decreased, and the accuracy of point cloud data acquisition was improved;
- (2) The density clustering algorithm was improved by the linear decreasing particle swarm optimization (PSO) algorithm. In order to solve the problem of sparse density, in this paper, a pre-trained PU-net was used for enhancing the point cloud density of the point cloud data obtained by UAV-borne LiDAR. Experiments have verified that the enhanced point cloud data retain the features of the raw point cloud data.

2. Framework of the Proposed Method

For UAV-borne LIDAR point cloud data enhancement, the integrated navigation algorithm based on GOA and the point cloud density enhancement algorithm based on neural network were shown in Figure 1. The specific implementation steps were as follows:

- (1) Firstly, the GPS/INS integrated navigation information fusion framework was built by the Kalman filter. Then, the GOA was used to optimize the initial parameters of the Kalman filter, and the optimal solution was returned to the Kalman filter to estimate the output state of the GPS/INS integrated navigation system;
- (2) After data acquisition, the point cloud data was transformed into the WGS-84 coordinate system from the LiDAR coordinate system. Then the point cloud data was filtered for removing the irrelevant points. Finally, the two-dimensional point cloud data was transformed into a three-dimensional point cloud according to the flight trajectory of UAV;
- (3) The PSO algorithm was employed to optimize the density clustering algorithm for crop point cloud segment. Then the crop point cloud data was fed into the trained network to obtain the crop point cloud data after the point cloud density was enhanced. Finally, the point cloud data was reconstructed in three dimensions.

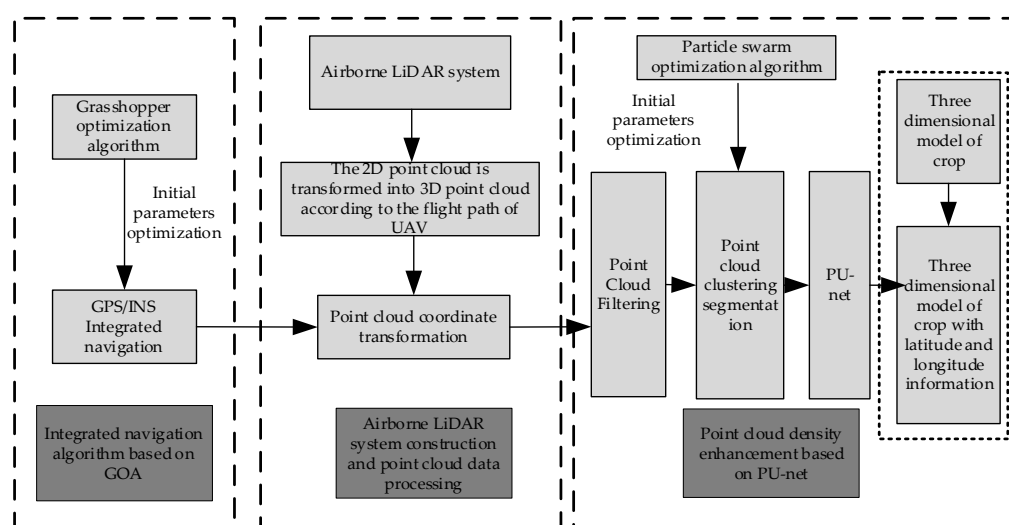


Figure 1. Framework of the proposed method. There are 3 steps: Integrated navigation algorithm based on grasshopper optimization algorithm (GOA), Unmanned Aerial Vehicle (UAV)-borne Light

Detection and Ranging (LiDAR) system construction and point cloud data processing, and point cloud density enhancement based on the Point Cloud Up-Sampling Network of the whole work of this paper.

3. Integrated Navigation Enhancement Based on GOA

In GPS/INS integrated navigation, the Kalman filter was usually used for fusing information of GPS and INS. The better the filter of integrated navigation was, the better it can integrate the advantages of each sensor. Since the GOA is not affected by the nonlinearity or scale, compared with other global optimization algorithms, it can find more effective optimal solutions with faster convergence speed [24].

3.1. Framework of GPS/INS Integrated Navigation

GPS and INS have complementary advantages. By combining GPS and INS in an appropriate way, their own drawbacks could be overcome, and the navigation accuracy would significantly improve. In order to improve the robustness of the system and reduce the complexity of the system, this paper adopted the loose combination mode to fuse the information of GPS and INS, as shown in Figure 2.

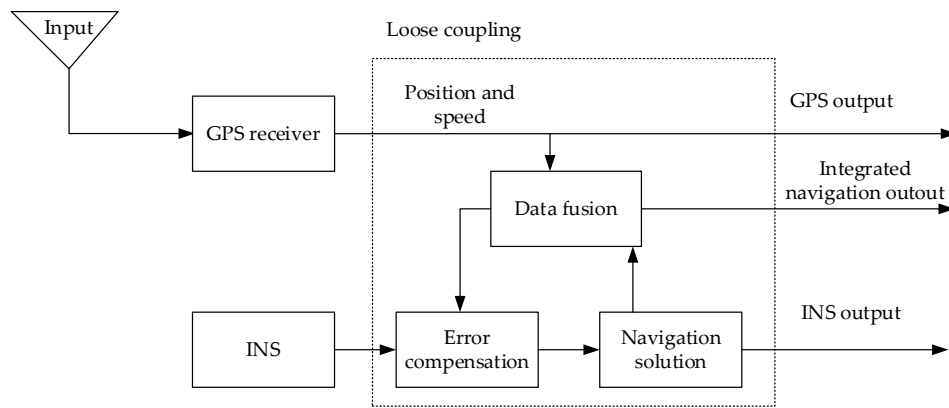


Figure 2. Framework of global positioning system (GPS)/ inertial navigation system (INS) integrated navigation.

The state variables of the integrated navigation filter were composed of the random drift error, velocity error, and position error of INS gyroscope, as well as errors caused by the attitude angle and accelerometer. The equation of the state was as follows:

$$\dot{\hat{X}} = F_X \hat{X} + G_W \quad (1)$$

The state variables of the INS integrated navigation system include longitude error, latitude error, velocity error in east and north directions, attitude angle error in east, north, and sky directions, random drift, and accelerometer bias error. The 13-dimensional equation of the state was as follows:

$$\dot{\hat{X}}_I = F_I \hat{X}_I + G_I \omega_I \quad (2)$$

The clock error of the receiver equivalent distance δt_u and equivalent range error of the receiver clock frequency error δt_{ru} were the GPS state errors; the equation of the state can be described as follows:

$$\dot{\hat{X}}_G = F_G \hat{X}_G + G_G w_G \quad (3)$$

The measurement equation of the position velocity combination was expressed as follows:

$$\begin{bmatrix} L_{INS} \\ \lambda_{INS} \\ h_{INS} \end{bmatrix} = \begin{bmatrix} L_t + \delta L \\ \lambda_t + \delta \lambda \\ h_t + \delta h \end{bmatrix} \quad (4)$$

where L_{INS} , λ_{INS} , and h_{INS} were the longitude, latitude, and altitude calculated by inertial navigation. L_t , λ_t , and h_t were the real longitude, latitude, and height of the carrier. δL , $\delta \lambda$, and δh were the longitude and latitude error and altitude error calculated by INS.

The position measurement information of the GPS receiver was expressed by subtracting the corresponding error from the real value:

$$\begin{bmatrix} L_{GPS} \\ \lambda_{GPS} \\ h_{GPS} \end{bmatrix} = \begin{bmatrix} L_t - \frac{N_n}{R} \\ \lambda_t - \frac{N_e}{R \cos L} \\ h_t - N_u \end{bmatrix} \quad (5)$$

where N_n , N_e , and N_u were the position error of the GNSS receiver along the east, north, and sky directions. Then the external measurement position error can be defined as:

$$Z_{pos} = \begin{bmatrix} \lambda_{INS} - \lambda_{GPS} \\ L_{INS} - L_{GPS} \\ h_{INS} - h_{GPS} \end{bmatrix} \quad (6)$$

The position measurement equation was as follows:

$$Z_{POS} = H_{POS}(t)X(t) + V_{POS}(t) \quad (7)$$

The velocity measurement information equation of INS can be expressed as follows:

$$\begin{bmatrix} v_{eINS} \\ v_{nINS} \\ v_{uINS} \end{bmatrix} = \begin{bmatrix} v_e + \delta v_e \\ v_n + \delta v_n \\ v_u + \delta v_u \end{bmatrix} \quad (8)$$

where v_{eINS} , v_{nINS} , and v_{uINS} were the east speed, north speed, and sky speed calculated by the inertial navigation system. v_e , v_n , and v_u were the velocity of the object in the east, north, and sky directions.

The velocity information provided by GPS can be expressed by the difference between the real value and the corresponding error in the geographical coordinate system:

$$\begin{bmatrix} v_{eGPS} \\ v_{nGPS} \\ v_{uGPS} \end{bmatrix} = \begin{bmatrix} v_e - M_e \\ v_n - M_n \\ v_u - M_u \end{bmatrix} \quad (9)$$

The speed error of the external measurement can be defined as:

$$Z_{vel} = \begin{bmatrix} v_{eINS} - v_{eGPS} \\ v_{nINS} - v_{nGPS} \\ v_{uINS} - v_{uGPS} \end{bmatrix} \quad (10)$$

The velocity measurement equation was as follows:

$$Z_{vel} = H_{vel}(t)X(t) + V_{vel}(t) \quad (11)$$

The integrated measurement equation of the GPS/INS integrated navigation system was obtained as follows:

$$Z(t) = \begin{bmatrix} \lambda_{INS} - \lambda_{GPS} \\ L_{INS} - L_{GPS} \\ h_{INS} - h_{GPS} \\ v_{eINS} - v_{eGPS} \\ v_{nINS} - v_{nGPS} \\ v_{uINS} - v_{uGPS} \end{bmatrix} = H(t)X(t) + V(t) \quad (12)$$

3.2. Grasshopper Optimization Algorithm (GOA)

GOA is a swarm intelligence optimization algorithm designed by Australian scholar Shahrzad [18] in 2017, simulating the predatory migration behaviors of the grasshopper population in nature. The main characteristics of larval stage grasshoppers were slow movement and small stride. In contrast, long distances and sudden movements were essential features of adult grasshoppers. Searching for food sources was another important feature of the grasshopper swarm. The mathematical model of grasshopper swarm behavior was as follows:

$$X_i = \sum_{\substack{j=1 \\ j \neq i}}^N s(|x_j - x_i|) \frac{x_j - x_i}{d_{ij}} - g\hat{e}_g + u\hat{e}_w \quad (13)$$

where X_i represented the position of i_{th} grasshopper. N represents the amount of the grasshopper swarm. s was interaction force between grasshoppers. d_{ij} was the distance between the i_{th} grasshopper and j_{th} grasshopper. The function of the interaction force (attraction and repulsion shown in Figure 3.) s between grasshoppers was defined as follows:

$$s(d) = fe^{\frac{-d}{l}} - e^{-d} \quad (14)$$

where d was the distance between two grasshoppers. f represented the interaction force (attraction and repulsion), l represented the unit of interaction force length. The modified form of Equation (10) was as follows:

$$X_i^{dim} = c * r \left(\sum_{\substack{j=1 \\ j \neq i}}^N c \frac{ub_{dim} - lb_{dim}}{2} A \frac{x_j - x_i}{d_{ij}} \right) + \hat{T}_{dim} \quad (15)$$

$$A = s(|x_j^d - x_i^d|) \quad (16)$$

where ub_{dim} was the upper boundary of the dim_{th} dimension, lb_{dim} was the lower boundary, and $\hat{T}_{dim}$ was the target value of the dim_{th} dimension. c represented a self-adaption ratio which is defined as follows:

$$c = c_{max} - l \frac{c_{max} - c_{min}}{L} \quad (17)$$

where c_{max} was the max value, while c_{min} was the minimum value; l represented the current iteration, while L is the max iteration.

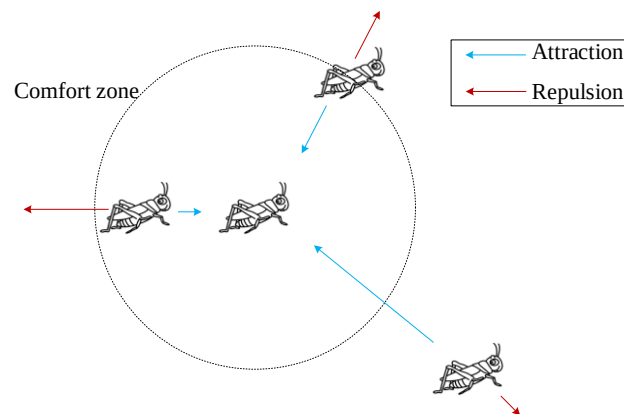


Figure 3. Interaction among grasshoppers.

3.3. Kalman Filter Optimization Using GOA

GOA was employed to optimize the initial parameters Q and R of the Kalman filter. Q was the variance matrix of the system noise sequence, which affected the filtering performance and parameter estimation accuracy of the Kalman filter algorithm; R was the variance matrix of the measurement noise sequence, which was related to the correction speed of filtering and the stability of the filtering process. Through the optimization of the initial parameters, the optimal position of the grasshopper was as close to the real value as possible, thus the accuracy of the Kalman filter algorithm was improved greatly. The sum of the variances of longitude and latitude errors was selected as the fitness function. The pipelines of the Kalman filter optimization was shown in Figure 4.

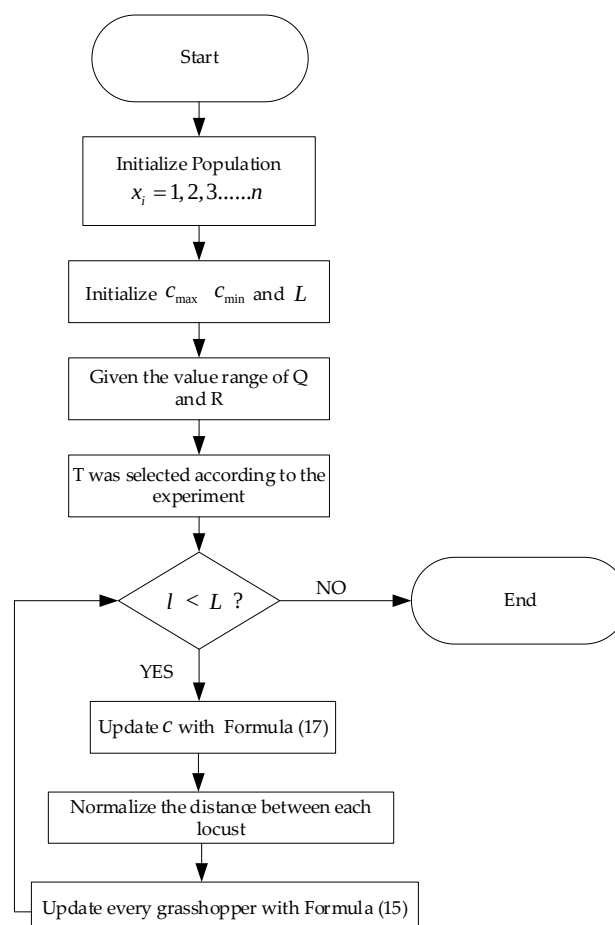


Figure 4. The pipelines of Kalman filter optimization.

4. UAV-Borne LiDAR System Construction and Point Cloud Data Processing

The UAV-borne LiDAR system was mainly composed of the LiDAR sensor, GPS/INS integrated navigation module, and UAV carrier. After the point cloud data were collected by the UAV-borne LiDAR scanning system, it was necessary to transform the point cloud data from the LiDAR coordinate system to the WGS-84 coordinate system.

4.1. UAV-Borne LiDAR System

In view of the application of LiDAR in the agricultural field, this paper selected RPLiDAR-a2 of SLAMTEC®, which was a single-line two-dimensional LiDAR, which greatly reduced the cost of equipment. The WitMotion® GPS/ Inertial Measurement Unit (IMU) was selected to collect the navigation and positioning information of UAV. In order to supply electricity to the LiDAR and the IMU, a small pad was used to combine the system. DJI® M100 UAV was selected as the carrier, which was a lightweight UAV. Finally, the UAV-borne LiDAR scanning system was completed, as shown in Figures 5 and 6.

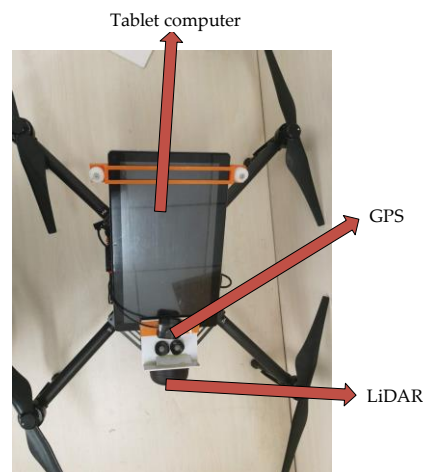


Figure 5. Top view of UAV-borne LiDAR scanning system.

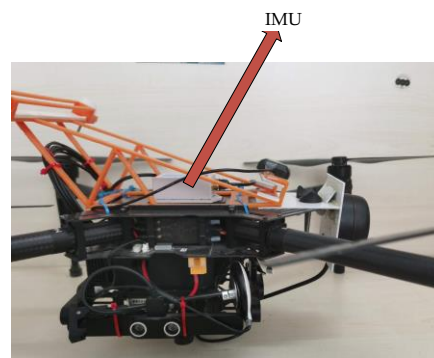


Figure 6. Side view of UAV-borne LiDAR system.

Due to the single scanning line of RPLiDAR-a2, the collected point cloud data was a two-dimensional point cloud, which only contained the data of Y and Z coordinates. In order to generate three-dimensional point cloud data, according to the light trajectory and speed of UAV, the data of the X-axis were inversely solved according to IMU.

4.2. Coordinate Transformation of Point Cloud Data

It was assumed that the coordinate of the laser foot point in the instantaneous laser coordinate system was $(x_{SL}, y_{SL}, z_{SL})^T$:

$$\begin{bmatrix} x_{SL} \\ y_{SL} \\ z_{SL} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \rho \end{bmatrix} \quad (18)$$

There was an angle θ between the instantaneous laser coordinate system and the laser scanning reference coordinate system. The coordinates of the raw point cloud data in the laser scanning reference coordinate system were $(x_L, y_L, z_L)^T$:

$$\begin{bmatrix} x_L \\ y_L \\ z_L \end{bmatrix} = R_L \begin{bmatrix} x_{SL} \\ y_{SL} \\ z_{SL} \end{bmatrix} \quad (19)$$

where R_L represented the rotation matrix of scan angle θ .

Generally, the error between the different coordinate axes was analyzed and unified to be the placement error angle α , β , and γ . The deviation of the coordinate origin and reference center was treated as error $t_L = (\Delta x_I^L, \Delta y_I^L, \Delta z_I^L)^T$. The coordinates of the raw point cloud in the inertial platform reference coordinate system were $(x_I, y_I, z_I)^T$:

$$\begin{bmatrix} x_I \\ y_I \\ z_I \end{bmatrix} = R_M \begin{bmatrix} x_L \\ y_L \\ z_L \end{bmatrix} + \begin{bmatrix} \Delta x_I^L \\ \Delta y_I^L \\ \Delta z_I^L \end{bmatrix} \quad (20)$$

where $R_M = R(\gamma) \times R(\beta) \times R(\alpha)$.

In the integrated navigation system, there were also positioning errors between GPS and INS. During the installation process, there was a certain deviation between the center of GPS and the reference center of inertial components. The deviation was defined as $t_G = (\Delta x_I^G, \Delta y_I^G, \Delta z_I^G)^T$; it can be measured offline. INS measured the attitude information of the carrier in real time, including roll angle (R), pitch angle (P), and heading angle (H), and got the velocity and position information of the carrier through integration and feed back to the user in real time. At the same time, the coordinate rotation matrix was obtained by calculating the attitude information feedback from IMU. The coordinates of the raw point cloud data in the local reference coordinate system were $(x_{LH}, y_{LH}, z_{LH})^T$:

$$\begin{bmatrix} x_{LH} \\ y_{LH} \\ z_{LH} \end{bmatrix} = R_N \begin{bmatrix} x_I \\ y_I \\ z_I \end{bmatrix} - \begin{bmatrix} \Delta x_I^G \\ \Delta y_I^G \\ \Delta z_I^G \end{bmatrix} \quad (21)$$

where $R_N = R(H) \times R(P) \times R(R)$.

Finally, point cloud data should be transformed to WGS-84 coordinate system. The rotation matrix R_W was a function of the latitude and longitude of the region. The point cloud could be transformed into the WGS-84 coordinate system by rotating the matrix:

$$\begin{bmatrix} x_{84} \\ y_{84} \\ z_{84} \end{bmatrix} = R_W \begin{bmatrix} x_{LH} \\ y_{LH} \\ z_{LH} \end{bmatrix} + \begin{bmatrix} x_{84} \\ y_{84} \\ z_{84} \end{bmatrix}_{\text{Antenna phase center}} \quad (22)$$

Based on the above analysis, the coordinates of the raw point cloud in WGS-84 coordinate system were as follows:

$$\begin{bmatrix} x_{84} \\ y_{84} \\ z_{84} \end{bmatrix} = R_W R_N \left[R_M R_L \begin{bmatrix} 0 \\ 0 \\ \rho \end{bmatrix} + \begin{bmatrix} \Delta x_I^L \\ \Delta y_I^L \\ \Delta z_I^L \end{bmatrix} - \begin{bmatrix} \Delta x_I^G \\ \Delta y_I^G \\ \Delta z_I^G \end{bmatrix} \right] + \begin{bmatrix} x_{84} \\ y_{84} \\ z_{84} \end{bmatrix}_{\text{Antenna phase center}} \quad (23)$$

In addition, in order to get the data containing the latitude and longitude information, it was necessary to convert the three-dimensional rectangular coordinates (XYZ) into geodetic coordinates (λ LH). The conversion formula was as follows:

$$L = \arctan(Y/X) \quad (24)$$

$$\lambda = \arctan \left[Z / \sqrt{X^2 + Y^2} * (1 - e^2 N(N + H))^{-1} \right] \quad (25)$$

$$H = \sqrt{X^2 + Y^2} / \cos \lambda - N \quad (26)$$

where X , Y , and Z were three-dimensional rectangular coordinate components, L , λ , and H were earth longitude, latitude, and height. e represented the first eccentricity of the ellipsoid. $N = a / \sqrt{1 - e^2 \sin^2 \lambda}$ was the radius of curvature in the prime vertical, and the a was the semimajor axis of the ellipsoid.

5. Point Cloud Density Enhancement Based on PU-Net

Point cloud density enhancement aims to increase the number of point clouds without destroying the characteristics of the original point cloud. By enhancing the density of point cloud, the richness and accuracy of 3D reconstruction of point cloud data can be improved without increasing equipment cost.

5.1. Point Cloud Preprocessing

After the coordinate transformation, it was necessary to remove irrelevant noise points from the original point cloud data. In this paper, the statistical filtering method was selected for filtering. Statistical filtering calculated the distance between each point and its surrounding neighborhood points: According to neighborhood points and the standard deviation threshold, the points which did not meet the requirements were identified as outlier points.

5.2. Point Cloud Density Clustering Segmentation Method Based on PSO Algorithm

After filtering in the previous section, the point cloud data obtained included ground and crop points. In order to separate the ground and crop, the density clustering algorithm was used for separating, and the linear decreasing PSO algorithm was employed to improve the density clustering algorithm.

DBSCAN was a density-based clustering algorithm proposed by Ester et al. It was designed to achieve clustering and segmentation of arbitrary data in spatial and non-spatial high-dimensional databases. By calculating the distance between each point and its neighborhood, the points with similar density were classified into the same class.

ϵ -Neighborhood of arbitrary point p is defined as follows:

$$N_{\epsilon p} = \{q \in D / \text{dist}(p, q) < \epsilon\} \quad (27)$$

where D represented the target's database, point p was the core point if the ϵ -neighborhood contained at least the minimum number of points. The core point was defined as follows:

$$N_{\epsilon p}(P) > \text{MinPts} \quad (28)$$

where Eps and MinPts were the neighborhood radius and minimum number of points in the ε -neighborhood of the core point. Through the calculation of all the points, the density of the points meeting the set conditions was similar, and they were classified as the same kind of points. All points were calculated until the termination condition was met. Because the selection of these two initial parameters mainly depended on human experience and continuous experiments, in order to improve the efficiency, this paper employed the linear decreasing PSO algorithm to iteratively optimize the two initial parameters to determine the optimal solution of the initial parameters.

The PSO algorithm was a swarm intelligent bionic optimization algorithm: Every point in PSO was stamped by a position P_i , the optimal value of swarm was P_g , every point's velocity was V_i , and each particle's velocity and position update formula was shown as follows:

$$V_i^{t+1} = \omega V_i^t + c_1 r_1 (P_i - X_i^t) + c_2 r_2 (P_g - X_i^t) \quad (29)$$

$$X_i^{t+1} = X_i^t + V_i^{t+1} \quad (30)$$

where t represented the iterations, V_i^{t+1} was the velocity of the $t + 1$ iteration, ω represented the inertia weight. $c_1, c_2 \in [0, 1]$ were the generating functions of a random number, and X_i^{t+1} was the $t + 1$ iteration position.

In the initial stage of the algorithm, the inertia weight ω was given a larger value to enhance the global exploration ability of the algorithm. With the continuous exploration process, the value of ω could be linearly reduced to enhance the local search and optimization ability of the algorithm, so that particles could gradually approach the optimal solution in the neighborhood of the optimal value. The value of ω was expressed as follows:

$$\omega = \omega_{\min} + \frac{(\omega_{\max} - \omega_{\min}) \times (\max_{\text{iter}} - t)}{\max_{\text{iter}}} \quad (31)$$

where $\omega_{\max}$ represented the max inertia coefficient, while $\omega_{\min}$ represented the minimum. $\max_{\text{iter}}$ was the max iteration.

PSO optimization pipelines are shown in Figure 7:

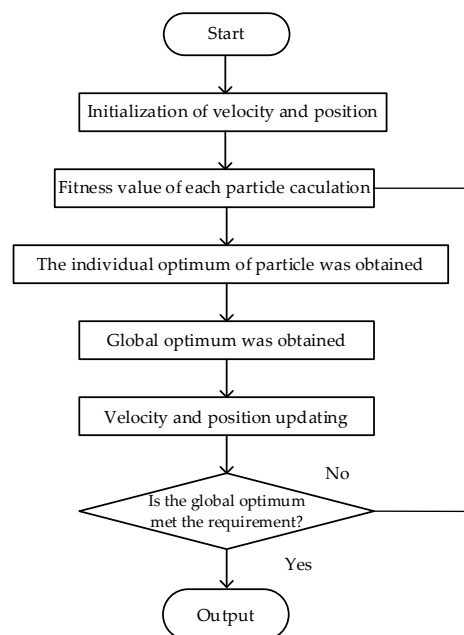


Figure 7. Flow chart of particle swarm optimization (PSO) algorithm.

5.3. Point Cloud Density

The point cloud data acquired by UAV-borne LiDAR was 3-dimensional, which was usually expressed by plane density. The point cloud density in national standards was defined as the normal direction of elevation density [22]:

$$\rho = \frac{N - \sum_{i=0}^k N_i}{A - \sum_{i=0}^k A_i} \quad (32)$$

where ρ represented the point cloud density, and N represented the total amount of points. Furthermore, N_i was the i_{th} field number of points, and k was the number of scanning area. A was the measure of the scanning area.

The point cloud density calculated by the above formula was the average point cloud density in the scanning area. In order to further quantify the point density distribution in the scanning area, the reciprocal of the area affected by a single point was taken as the density of the point:

$$d' = 1/A' \quad (33)$$

The main factors affecting the density of the point cloud were the LiDAR design parameters, system characteristics, carrier, and scanning area.

The distance between points directly reflected the density of point clouds. For the pulse laser ranging sensor, the distance from the heading point to the edge point determined the density distribution of point cloud. The design of aerial photography parameters was referred from the distance between heading point and edge point. The parameters of the sensor, such as scanning frequency, scanning bandwidth, laser emission frequency, field of view angle, flight height and flight speed, and the distance between point clouds, were considered to design the LiDAR system.

5.4. Density Enhancement Using Point Cloud Up-Sampling Network (PU-Net)

Point cloud up-sampling was essentially similar to the problem of image super-resolution [25]. Yu et al. [26] proposed a data-driven point cloud sampling neural network (PU-net), which was applied to patch level. At the same time, a joint loss function was used for guiding the up-sampled points to retain a uniformly distributed object surface. The main idea of PU-net was capturing the features of each point in every level, then multi-convolution blocks were extended in feature hidden layers, and finally, the hidden layers were divided into multi-level features and up-sampled into the point cloud. The pre-trained PU-net was employed for enhancing the density of the crop point cloud in this paper. There were 4 steps of PU-net for up-sampling of point cloud: Slice extraction, point feature embedding, feature extending, and point coordinate reconstruction. It is noteworthy that a series of sub-pixel convolution layers were employed to extend the features of neighborhood points in the feature extending stage:

$$f' = \text{Reshape}([\text{conv}_1^2(\text{conv}_1^1(f)), \dots, \text{conv}_r^2(\text{conv}_r^1(f))]) \quad (34)$$

where $\text{conv}_i^1(*)$ and $\text{conv}_i^2(*)$ were the convolution layer with 1×1 kernel size for adjusting the channels of patch feature f . The output, combined feature f' , was born with properties of spatial characteristic which could retain the information of raw data. After coordinate reconstruction, the density elevation in last section was employed to show the enhancement effect directly.

6. Results

In order to verify the stability of the UAV-borne LiDAR scanning system, the experiments were carried out in integrated navigation testing, point cloud density clustering, and point cloud density enhancement.

6.1. GOA-Based Integrated Navigation

The integrated navigation experiment was carried out in MATLAB2017 (a). It was supposed that the initial longitude of UAV $L = 40^\circ$, the initial latitude $\lambda = 116^\circ$, and the velocity of UAV was 1m/s. The filtering period was $T = 1s$. The initial value of filtering covariance was as follows:

$$p(0) = \text{diag}[(0.00254^\circ)^2, (0.00446^\circ)^2, (0.1\text{m/s})^2, (0.1\text{m/s})^2, (1^\circ)^2, (1^\circ)^2, (1^\circ)^2, (1^\circ)^2, (1^\circ)^2, (1^\circ)^2, (10^{-5}g), (10^{-5}g), (10^{-5}g)] \quad (35)$$

The initial value of integrated navigation was:

$$X(0) = [0\text{m}, 0\text{m}, 0.5\text{m/s}, 0.5\text{m/s}, 0.08^\circ, 0.08^\circ, 0.08^\circ, 0.16^\circ, 0.1^\circ/\text{h}, 0.1^\circ/\text{h}, 0.1^\circ/\text{h}, 0.1^\circ/\text{h}, (10^{-5}g), (10^{-5}g), (10^{-5}g)] \quad (36)$$

The initial parameters Q and R obtained when the sum of longitude and latitude errors were used as the fitness function feeding into the system. The GOA based integrated navigation error diagram was shown in Figures 8 and 9, PSO based integrated navigation error diagram was shown in Figures 10 and 11.

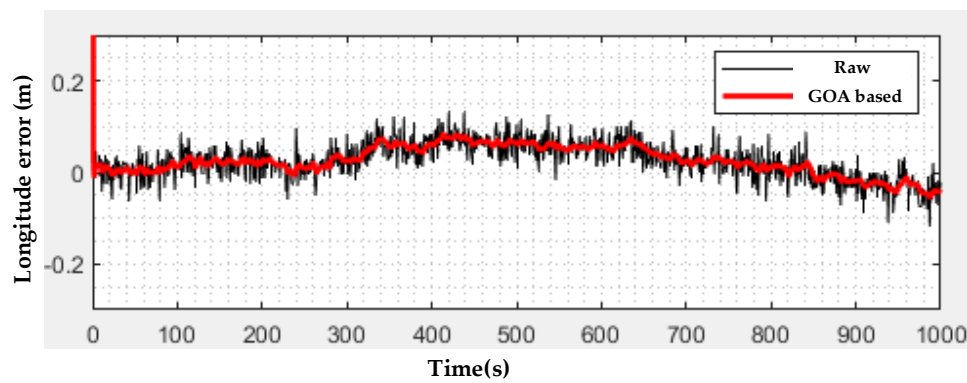


Figure 8. Longitude error of GOA-based integrated navigation.

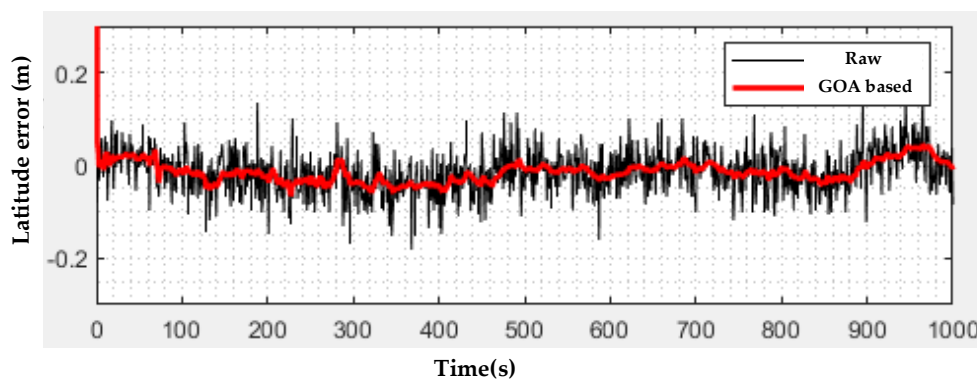


Figure 9. Latitude error of GOA-based integrated navigation.

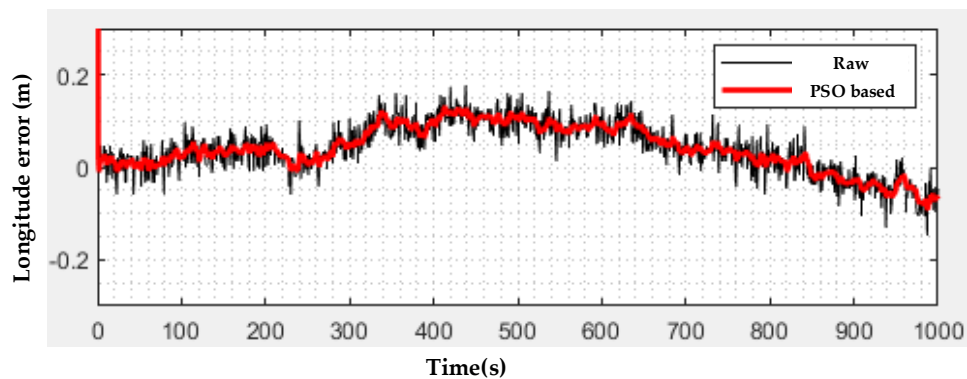


Figure 10. Longitude error of PSO-based integrated navigation.

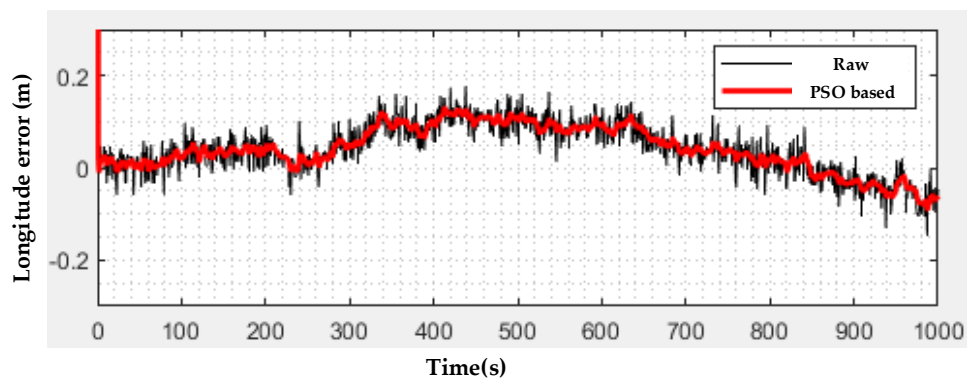


Figure 11. Latitude error of PSO-based integrated navigation.

As shown in Table 1, both the longitude L variance error and latitude λ variance error were reduced greatly compared with raw integrated navigation. It is worth noting that the GOA-based integrated navigation showed better performance than the PSO-based integrated navigation; it is suspected that the ability to find the optimal of GOA was better than for PSO in the integrated navigation task. For more precise optimal Q and R , the GOA-based integrated navigation was employed for the UAV-borne LiDAR scanning system.

Table 1. Error of integrated navigation with different enhancement methods.

	Longitude L Variance Error	Latitude λ Variance Error
GOA-based integrated navigation	0.00046	0.00034
PSO-based integrated navigation	0.00087	0.00079
Raw integrated navigation	0.0091	0.0047

6.2. PSO-Based Point Cloud Clustering Segmentation

The meta-heuristic algorithm was employed to optimizing the clustering segmentation key elements: Eps and MinPts. The Davies–Bouldin index (DBI) was employed as the fitness function for the joint optimization of Eps and MinPts:

$$DBI = \frac{1}{k} \sum_{i=1}^k \max_{j \neq i} \left(\frac{\text{avg}(C_i) + \text{avg}(C_j)}{d_{\text{cen}}(\mu_i, \mu_j)} \right) \quad (37)$$

where $\text{avg}(\cdot)$ represented the average distance between samples in cluster, $d_{\text{cen}}(\mu_i, \mu_j)$ was the center distance of cluster μ_i , and cluster μ_j . $\text{avg}(\cdot)$ was defined as follows:

$$\text{avg}(C) = \frac{2}{|C|(|C| - 1)} \sum_{1 \leq i < j \leq |C|} \text{dist}(x_i, x_j) \quad (38)$$

where $\text{dist}(x_i, x_j)$ was the distance of sample x_i and sample x_j .

Crop point cloud data collection environment was shown in Figure 12.



Figure 12. Crop point cloud raw data collection environment.

A series of crop point cloud raw data was collected from the UAV-borne LiDAR scanning system; the meta-heuristic-based DBSCAN algorithms were initialized by crop raw data and iterated 200 times to find the optimal Eps and MinPts. The fitness iteration curve is shown in Figure 13.

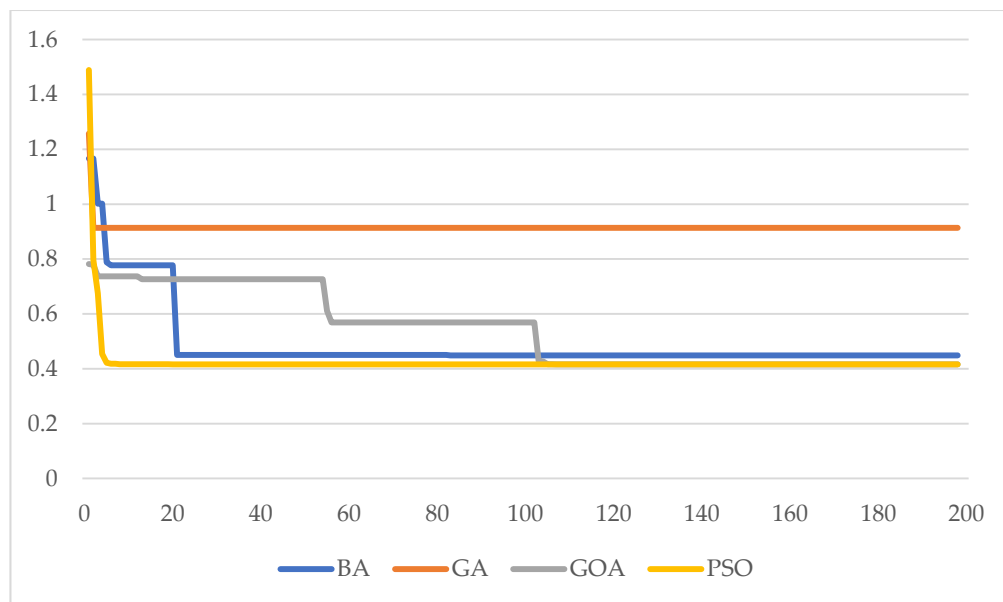


Figure 13. Iteration curve of fitness of the meta-heuristic-based density-based spatial clustering of applications with noise (DBSCAN) algorithms.

For fair comparing with different meta-heuristic algorithms, the BA-based DBSCAN algorithm [17], GA-based DBSCAN algorithm [14], GOA-based DBSCAN algorithm [18], and PSO-based DBSCAN algorithm [15] were initialized with 20 population-sizes, 2 dimensions, and 200 iterations. In particular, the GA-based DBSCAN algorithm was initialized with a 0.7 crossover rate, 0.5 select rate, and 0.01 variation rate. During several times of testing, Figure 13 shows that the GOA and PSO found the

best fitness 0.416032. It is suspected that global optimization was 0.416032 under further testing with different fine tuning. Under the same fitness 0.416032, the Eps = 0.8147 and MinPts = 13. In the clustering segmentation, the BA-based DBSCAN algorithm, GA-based DBSCAN algorithm, GOA-based DBSCAN algorithm, PSO-based DBSCAN algorithm, and the K-means clustering were employed for comparison; the K-means was the default tuning. The 200 iterations of time consumption of the BA-based DBSCAN algorithm, GA-based DBSCAN algorithm, GOA-based DBSCAN algorithm, and PSO-based DBSCAN algorithm were 9669 s, 910 s, 1610 s, and 749 s, respectively. The comparison clustering results are shown in Figure 14a–e. Shown in Figure 14f, the crop point cloud is reconstructed in 3-dimensions.

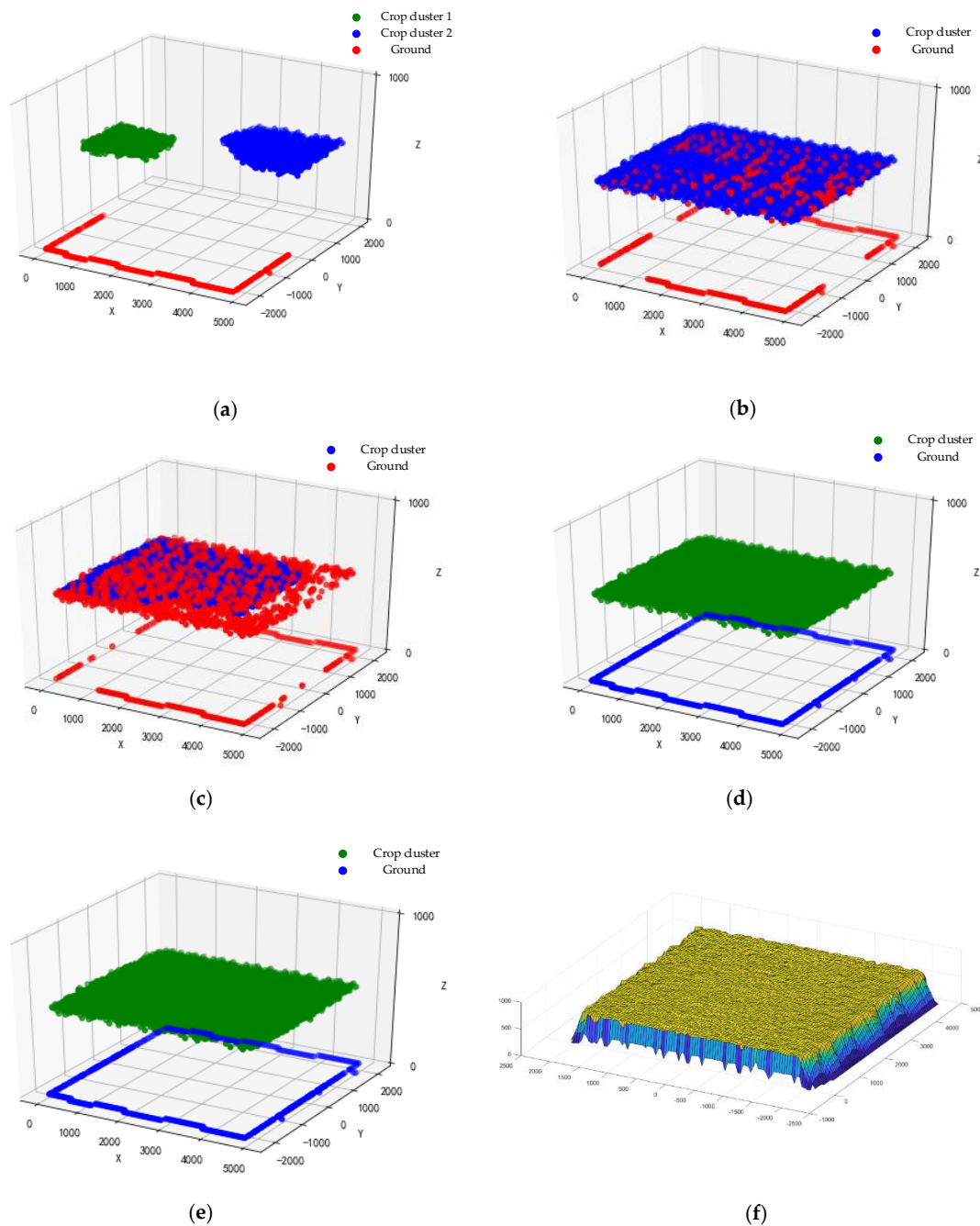


Figure 14. Clustering segmentation result of crop and ground. (a) K-means clustering; (b) BA-based DBSCAN (Eps = 0.2145, MinPts = 18); (c) GA-based DBSCAN (Eps = 0.1006, MinPts = 8); (d) PSO-based DBSCAN (Eps = 0.8147, MinPts = 13); (e) GOA-based DBSCAN (Eps = 0.8147, MinPts = 13); (f) 3D reconstruction of crop field point cloud.

As shown in Figure 14a–c, 3 kinds of clustering results were given from the default tuning of K-means, and the ground cluster could not be separated from the crop cluster using the BA-based DBSCAN algorithm or GA-based DBSCAN algorithm. Under the same fitness 0.416032, the Eps = 0.8147 and MinPts = 13 were obtained using the GOA-based DBSCAN algorithm and PSO-based DBSCAN algorithm. Considering that PSO converges faster and takes less time in iterations, the PSO-based DBSCAN algorithm was employed for crop point cloud clustering segmentation.

6.3. Density Enhancement with PU-Net

After the clustering segmentation of crop point cloud, the crop point cloud was enhanced by a pre-trained PU-net. The PU-net was trained with 60 sets of variation of point clouds from “Visionair”. Datasets were divided into train sets and validation sets by 60% and 40%. In the fine-tune training period, the Monte Carlo stochastic sampling method was employed for sampling 5000 points from every object as the input. Each input represented the global information of each object; fine-tune training aimed at enhancing the global information collection performance of the pretrained model. The visualization of raw crop point cloud data was shown in Figure 15. The sparse part of the point cloud was mainly concentrated on the X-axis, which was the direction of the UAV-borne LiDAR scanning system to generate the 3D point cloud by moving forward.

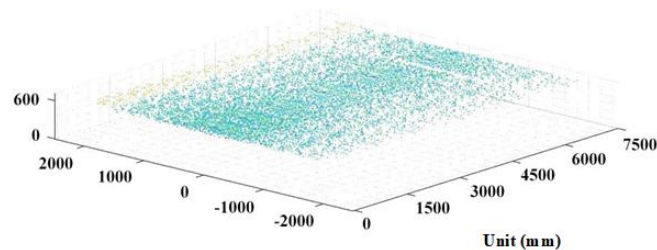


Figure 15. Visualization of raw crop point cloud data.

The raw crop point cloud was fed into the fine-tune trained PU-net. The 3D reconstruction of the crop point cloud with and without density enhancement is shown in Figure 16.

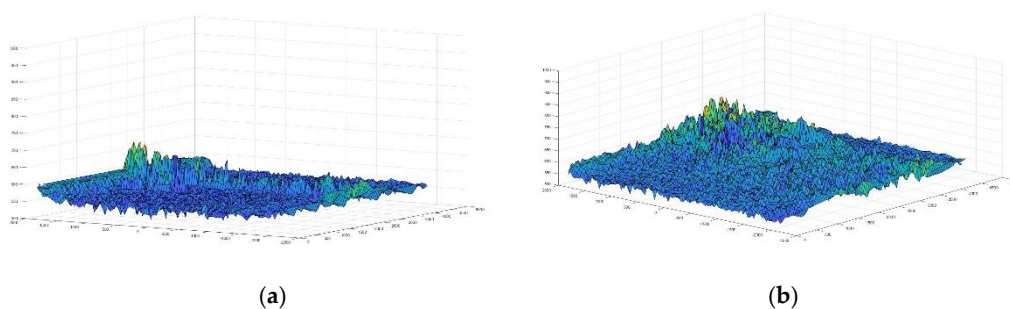


Figure 16. 3D reconstruction of crop point cloud with and without density enhancement. (a) The raw crop point cloud; (b) The crop point cloud with density enhancement.

The total number of points of the raw crop point cloud were 10,000, the density of the raw crop point cloud was 1250 points per square meter. After density enhancement, the total number of points of the crop point cloud were 40,000, the density of the crop point cloud was 5000 points per square meter, and the number and density of the point cloud had been significantly increased. Compared with the raw crop point cloud, the enhanced crop point cloud contained more information, and also retained the shape and contour characteristics of the raw data. In order to further verify the reliability and effectiveness of the method, 50 sets of density enhancement tests were carried out and 5 sets of results were sampled randomly for statistical tests. The results were shown in Table 2.

Table 2. The statistical parameters of the enhanced point cloud and the raw data.

	Mean Value	Variance	Standard Deviation
Raw crop point cloud	558.64	171.60	13.10
1st set of the enhanced crop point cloud	568.84	298.48	17.27
2nd set of the enhanced crop point cloud	578.68	289.34	17.01
3rd set of the enhanced crop point cloud	554.32	278.56	16.69
4th set of the enhanced crop point cloud	565.44	325.08	18.03
5th set of the enhanced crop point cloud	563.56	317.51	17.82

7. Discussion

For integrated navigation testing, because the GOA and PSO showed better performance in fitness convergence velocity in the local field refer to Ref. [19]. Although this was an unfavorable feature for global optimization, for time varying and unpredictable problems, such as the integrated navigation system in this paper, better performance in the fitness convergence velocity in the local field tended to find the suboptimal solution quickly, which would reduce the error sharply. Shown as the GOA based integrated navigation error diagram (Figures 8 and 9) and the PSO based integrated navigation error diagram (Figures 10 and 11), both the longitude L variance error and latitude λ variance error were reduced greatly compared with raw integrated navigation. It is worth noting that the GOA-based integrated navigation showed better performance than the PSO-based integrated navigation; it is suspected that the ability to find the optimal of GOA was better than for PSO in the integrated navigation task. For more precise optimal Q and R , the GOA-based integrated navigation was employed for the UAV-borne LiDAR scanning system.

For point cloud density clustering segmentation, default-turning K-means clustering divided crop into two clusters and separates ground points clearly, but parts of data points were ignored for clustering. Different meta-heuristic-based DBSCAN algorithms shown different optimization capabilities according to Figure 13. GA algorithm was the weakest in point cloud density clustering, different variation rates and 800 iterations more were set in further fine-tuning, the GA-based DBSCAN algorithms still could not find the lower fitness value which was close to 0.416032. The BA-based DBSCAN algorithm shown the better optimization capability but lower computing efficiency, 9669 s of time consumption make it worse for practical application. Both GOA-based DBSCAN algorithm and PSO-based DBSCAN algorithm found the same fitness 0.416032. It is suspected that global optimization was 0.416032 under further testing with different fine tuning. Considering that PSO converges faster and takes less time in iterations, the PSO-based DBSCAN algorithm was employed for crop point cloud clustering segmentation.

For point cloud density enhancement, the pre-trained PU-net [26] was trained by “Visionair” datasets, there were many objects point cloud dataset for learning features by Convolutional Neural Network (CNN). After point cloud density enhancement, the density of point cloud was greatly enhanced, and Table 2 shown that there was no gap in mean value between the raw crop point cloud and 5 sets of the enhanced crop point cloud, which indicated that the enhanced point cloud kept the height distribution of the raw data. A faint change of variance and standard deviation indicated that the 5 sets of the enhanced point cloud also kept the shape and contour of the raw data.

8. Conclusions

This paper developed a UAV-borne LiDAR scanning system with enhanced integrated navigation and dense point cloud optimization. The GOA was employed for optimizing the integrated navigation; the longitude and latitude variances of optimized integrated navigation were 0.00046 and 0.00034, respectively, and the longitude and latitude variances of raw integrated navigation were 0.0091 and 0.0047, respectively. The positioning accuracy of the UAV carrier was improved greatly. The PSO was employed to find the optimal key elements of DBSCAN clustering segmentation, and accurate point cloud segmentation was obtained. At last, the fine-tuned PU-net was used for point cloud density

enhancement. Compared with the raw point cloud, the density of the enhanced crop point cloud was 4 times more, and the statistical parameters showed that the enhanced crop point cloud retained the shape and contour characteristics of the raw data. In this paper, each meta-heuristic optimization algorithm had its own merits for the different problems, but we still have not found an optimization algorithm that can meet all the processing requirements. In the future, owing to the flexibility of a multi-rotor UAV in remote sensing operations in specific areas, a multi-rotor UAV equipped with a small LiDAR or optical image remote sensing system will be more suitable for remote sensing operations in the agricultural field [27,28].

Author Contributions: Conceptualization, J.C. and Z.Z.; methodology, J.C. and Z.Z.; software, Z.Z. and K.Z.; validation, Z.Z., K.Z., and S.W.; formal analysis, S.W. and Y.H.; investigation, S.W. and Y.H.; data curation, K.Z. and S.W.; writing—original draft preparation, J.C. and Z.Z.; writing—review and editing, K.Z., S.W., and Y.H.; supervision, J.C. and Y.H.; project administration, J.C. and Y.H.; funding acquisition, J.C. and Y.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (Grant No. 51979275), by the National Key R&D Program of China (Grant Nos. 2017YFD0701003 from 2017YFD0701000, 2018YFD0700603 from 2018YFD0700600, and 2016YFD0200702 from 2016YFD0200700), by the Jilin Province Key R&D Plan Project (Grant Nos. 20180201036SF), by the Open Fund of Synergistic Innovation Center of Jiangsu Modern Agricultural Equipment and Technology, Jiangsu University (Grant No. 4091600015), by the Open Research Fund of State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University (Grant No. 19R06), by the Open Research Project of the State Key Laboratory of Industrial Control Technology, Zhejiang University (Grant No. ICT20021), and by the Chinese Universities Scientific Fund (Grant Nos. 2020TC033 and 10710301).

Conflicts of Interest: The authors declare no conflict of interest.

References

- McManamon, P.F. Review of LiDAR: A historic, yet emerging, sensor technology with rich phenomenology. *Opt. Eng.* **2012**, *51*, 060901. [\[CrossRef\]](#)
- Vázquez-Arellano, M.; Griepentrog, H.W.; Reiser, D.; Paraforos, D.S. 3-D Imaging Systems for Agricultural Applications—A Review. *Sensors* **2016**, *16*, 618. [\[CrossRef\]](#) [\[PubMed\]](#)
- Lovell, J.L.; Jupp, D.L.B.; Newnham, G.J. Measuring tree stem diameters using intensity profiles from ground-based scanning LiDAR from a fixed viewpoint. *ISPRS J. Photogramm. Remote Sens.* **2011**, *66*, 46–55. [\[CrossRef\]](#)
- Strahler, A.H.; Jupp, D.L.B.; Woodcock, C.E. Retrieval of forest structural parameters using a ground-based LiDAR instrument (Echidna). *Can. J. Remote Sens.* **2008**, *34*, 426–440. [\[CrossRef\]](#)
- Zheng, G.; Moskal, L.M.; Kim, S.H. Retrieval of effective leaf area index in heterogeneous forests with terrestrial laser scanning. *IEEE Trans. Geosci. Remote Sens.* **2013**, *51*, 777–786. [\[CrossRef\]](#)
- Lai, X.D.; Liu, Y.S.; Li, Y.X. Application status and development of point cloud density characteristics of airborne LiDAR. *Geospat. Inf.* **2018**, *16*, 1–5.
- Quan, W.; Gong, X.; Fang, J.; Li, J. Prospects of INS/CNS/GNSS Integrated Navigation Technology. In *INS/CNS/GNSS Integrated Navigation Technology*; Springer: Berlin/Heidelberg, Germany, 2015.
- Fu, B.; Liu, J.; Wang, Q. Multi-sensor integrated navigation system for ships based on adaptive Kalman filter. In Proceedings of the 2019 IEEE International Conference on Mechatronics and Automation (ICMA), Tianjin, China, 4–9 August 2019.
- Gao, Y.; Liu, S.; Atia, M.M.; Noureldin, A. INS/GPS/LiDAR integrated navigation system for urban and indoor environments using hybrid scan matching algorithm. *Sensors* **2015**, *15*, 23286–23302. [\[CrossRef\]](#) [\[PubMed\]](#)
- Shabani, M.; Gholami, A.; Davari, N. Asynchronous direct Kalman filtering approach for underwater integrated navigation system. *Nonlinear Dyn.* **2015**, *80*, 71–85. [\[CrossRef\]](#)
- Zhang, G.; Hsu, L.T. Intelligent GNSS/INS integrated navigation system for a commercial UAV flight control system. *Aerosp. Sci. Technol.* **2018**, *80*, 368–380. [\[CrossRef\]](#)
- Yu, H.; Han, H.; Wang, J.; Xiao, H.; Wang, C. Single-frequency GPS/BDS RTK and INS ambiguity resolution and positioning performance enhanced with positional polynomial fitting constraint. *Remote Sens.* **2020**, *12*, 2374. [\[CrossRef\]](#)

13. Norouz, M.; Ebrahimi, M.; Arbabmir, M. Modified Unscented Kalman Filter for improving the integrated navigation system. In Proceedings of the 25th Iranian Conference on Electrical Engineering, Tehran, Iran, 2–4 May 2017.
14. Goldberg, D. *Genetic Algorithms in Search, Optimization and Machine Learning*; Addison-Wesley: Boston, MA, USA, 1989.
15. Eberhart, R.; Kennedy, J. A new optimizer using particle swarm theory. In Proceedings of the 6th International Symposium on Micro Machine and Human Science, Nagoya, Japan, 4–6 October 1995; pp. 39–43.
16. Colomni, A.; Dorigo, M.; Maniezzo, V. Distributed optimization by ant colonies. In Proceedings of the 1st European Conference on Artificial Life, Paris, France, 11–13 December 1991; pp. 134–142.
17. Yang, X. A new metaheuristic bat-inspired algorithm. In *Proceedings of Nature Inspired Cooperative Strategies for Optimization (NISCO 2010)*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 65–74.
18. Saremi, S.; Mirjalili, S.; Lewis, A. Grasshopper optimisation algorithm: Theory and application. *Adv. Eng. Softw.* **2017**, *105*, 30–47. [[CrossRef](#)]
19. Feng, H.; Ni, H.; Zhao, R.; Zhu, X. An enhanced grasshopper optimization algorithm to the Bin packing problem. *J. Control Sci. Eng.* **2020**, *2020*, 3894987. [[CrossRef](#)]
20. Chen, Y.; Zhou, L.; Bouguila, N.; Wang, C.; Chen, Y.; Du, J. BLOCK-DBSCAN: Fast clustering for large scale data. *Pattern Recognit.* **2020**, *109*, 107624. [[CrossRef](#)]
21. Kedzierski, M.; Fryskowska, A. Methods of laser scanning point clouds integration in precise 3D building modelling. *Measurement* **2015**, *74*, 221–232. [[CrossRef](#)]
22. Rupink, B.; Mongus, D.; Zalik, B. Point density evaluation of airborne LiDAR datasets. *J. Univers. Comput. Sci.* **2015**, *21*, 587–603.
23. Huang, H.; Shi, W.U.; Gong, M. Edge-aware point set resampling. *ACM Trans. Graph.* **2013**, *32*, 1–12. [[CrossRef](#)]
24. Mafarja, I.A. Evolutionary population dynamics and grasshopper optimization approach for feature selection problems. *Knowl. Based Syst.* **2018**, *145*, 25–45. [[CrossRef](#)]
25. Ledig, C.; Theis, L.; Huszar, F. Photo-realistic single image super-resolution using generative adversarial network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
26. Yu, L.; Li, X.; Fu, C.W. PU-net: Point cloud upsampling network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 19–21 June 2018.
27. Zhang, Z.; Han, Y.; Chen, J.; Wang, S.; Wang, G.; Du, N. Information extraction of ecological canal system based on remote sensing data of unmanned aerial vehicle. *J. Drain. Irrig. Mach. Eng.* **2018**, *36*, 1006–1011.
28. Wang, S.; Han, Y.; Chen, J.; Pan, Y.; Cao, Y.; Meng, H. Weed classification of remote sensing ecological irrigation area by UAV based on deep learning. *J. Drain. Irrig. Mach. Eng.* **2018**, *11*, 1137–1141.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).