

Article

Construction and Optimization of a Collaborative Harvesting System for Multiple Robotic Arms and an End-Picker in a Trellised Pear Orchard Environment

Hewen Zhang ¹, Xiaoguang Li ¹, Ling Wang ², Dian Liu ² and Shubo Wang ^{1,*} 

¹ School of Automation, Institute of Intelligent Unmanned System, Qingdao University, Qingdao 266071, China; lixiaoguang@qdu.edu.cn (X.L.)

² College of Engineering, China Agricultural University, Beijing 100083, China; wangling2017068@cau.edu.cn (L.W.); sy20213071447@cau.edu.cn (D.L.)

* Correspondence: shubowang@qdu.edu.cn

Abstract: In order to meet the needs of intensive mechanized picking in trellised pear orchards, this paper designed a pick-place integrated end-picker based on the analysis of agronomic characteristics of trellised pear gardens and fruit. In order to realize the accurate positioning of pears in picking, based on the kinematic analysis of robot arms and the construction of a private dataset, the YOLOv5s object detection algorithm was used in conjunction with a depth camera to achieve fruit positioning. The hand–eye system calibration was carried out. Aiming at solving the problems of redundancy, inefficiency, and uneven distribution of task volume in the conventional multiple robot arms algorithm, a simulated annealing algorithm was introduced to optimize the picking sequence, and a task allocation method was proposed. On the basis of studying several key parameters affecting the performance of the algorithm, the picking efficiency was greatly optimized. And the effectiveness of the proposed multi-robot collaborative picking method in a trellised pear orchard environment was demonstrated through experiments and simulation verification. The experiments showed that the picking efficiency of the integrated end-picker was increased by about 30%, and the success rate was significantly higher than that of the flexible grippers. The results of this study can be utilized to advance robotic pear-picking research and development.

Keywords: fruit picking; multiple robotic arms; end-picker; target detection; task planning



Citation: Zhang, H.; Li, X.; Wang, L.; Liu, D.; Wang, S. Construction and Optimization of a Collaborative Harvesting System for Multiple Robotic Arms and an End-Picker in a Trellised Pear Orchard Environment. *Agronomy* **2024**, *14*, 80. <https://doi.org/10.3390/agronomy14010080>

Academic Editor: Baohua Zhang

Received: 1 December 2023

Revised: 25 December 2023

Accepted: 27 December 2023

Published: 28 December 2023



Copyright: © 2023 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 (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The trellised pear orchard, which originated in Japan, was introduced to China in the 1990s and was popularized rapidly. The trellised pear orchard has the following five advantages: good fruit quality; convenient operation and management; easy to implement standardized cultivation; being convenient for mechanized operation; and preventing wind and bird damage. The harvest time of pear, the target crop in the trellised orchard in this study, is affected by a variety of characteristics and climatic conditions, so it is very important to harvest fruits at the right time according to specific requirements. Since the efficiency of a single robot arm is low, it is not enough to meet the demand of fruit harvest, and thus this study used multiple-robot-arm cooperative picking to improve efficiency and avoid the problems of quality degradation and low picking efficiency caused by not picking in time.

In general, picking work is divided into three aspects: perception, decision making, and control execution. The environment and fruit information are perceived by sensors. The decision system judges the picking target, and the execution system is driven to complete the picking action. The whole system is known as the “hand–eye–brain” picking system. This research direction has had wide concern worldwide in recent years.

Visual perception target detection in fruit and vegetable picking research is mainly divided into one-stage detection algorithms and two-stage detection algorithms. One stage detection algorithms, such as YOLO, SSD, SqueezeDet, and DetectNet, can directly extract features from the network to predict the classification and position of objects, thus eliminating the need for regional candidate networks (PRNs). The outstanding feature is that the detection speed is fast and only requires one step. On the contrary, two-stage detection algorithms, such as RCNN, FasterR-CNN, and MaskR-CNN, need to first create a proposal box that may contain the object to be detected and then perform further detection based on the object characteristics to complete the recognition and positioning of the target, which is more accurate. Mu et al. [1] used FasterR-CNN to identify kiwifruit, wherein they input the image and depth image obtained by a Kinectv2 camera (Microsoft, Redmond, WA, USA) into the convolution neural network to detect and locate the kiwifruit in the picture. Yang et al. [2] proposed a citrus fruit and branch recognition model based on MaskRCNN for fruit recognition and location under different occlusion conditions, and they constructed training datasets under a variety of complex conditions, including single fruit, multiple fruits, covered fruits, branches, and trunks. Qian et al. [3] proposed a method for mushroom detection and location based on SSD that optimizes the backbone network of the original SSD model to improve the real-time detection performance in embedded devices. The model has good detection performance for *Pleurotus ostreatus*. For apple fruit and branch segmentation, Kang et al. [4] adopted the Dsnet network model. Peng et al. [5] used the DeepLabV3+ semantic segmentation model based on the Xception_65 feature extraction network to detect litchi fruit. The experimental results showed that the model had 0.765 MIoU, which is 0.144 higher than the original DeepLabV3+ model. In order to adapt to the complex growth environment of litchi and simultaneously detect and locate the fruit branches of multiple litchi clusters, Li et al. [6] proposed a semantic segmentation method based on Deeplabv3 to segment the fruit, branches, and background in RGB images. However, due to the large differences in agronomic characteristics between fruits, we need to develop semantic segmentation and a target detection algorithm for the trellised pear orchard scene on the basis of the above fruit recognition algorithm.

In order to meet the needs of recognition and location of litchi fruit and stem at night, Liang et al. [7] proposed a litchi fruit detection method based on YOLOv3. In order to verify whether different classification modes will affect the detection effect of the kiwifruit detection model, Suo et al. [8] collected and classified 1160 kiwifruit images according to picking strategy and occlusion conditions, and they inputted them into YOLOv4 and YOLOv3 network models for training and testing. The experimental results showed that the tagging and classification of datasets in a way that is as detailed as possible can effectively improve the detection accuracy of the network model. Xiong et al. [9] developed a faster and more accurate system for the real-time vision detection, tracking, and locating of strawberries by combining YOLOv4, DeepSORT, and color threshold. In view of the low accuracy and poor robustness of the traditional green pepper detection methods, Li et al. [10] proposed an improved green pepper target detection algorithm based on Yolov4_tiny. Based on the backbone network of the classical target detection model, the algorithm introduces adaptive feature fusion and feature attention mechanism to improve the accuracy of the small-target recognition of green pepper while ensuring the accuracy of classification. Aiming at the characteristics of the small size and dense growth of plums, Wang [11] proposed a lightweight model named improved YOLOv4, based on YOLOv4. The experiments showed that the improved YOLOv4 model had higher average accuracy (mAP); in addition, compared with YOLOv4, the size was compressed by 77.85%, and the detection speed was accelerated by 112%. Yan et al. [12] proposed an apple detection algorithm based on improved YOLOv5s that can effectively identify graspable apples and ungraspable apples, and the average detection time for a single image was found to be only 0.015 s. In order to meet the requirements of accuracy, lightweight model, and fast response during picking in a trellised pear orchard, on the basis of optimizing the YOLO

fruit detection algorithm, a pear detection algorithm based on improved YOLOv5s was developed, combined with a depth camera.

The terminal execution modes of agricultural picking robots usually include negative pressure adsorption, shear, mold cavity-sleeve, and flexible grasping. The Xiong team [13,14] of the Norwegian University of Life Sciences is devoted to the research of strawberry picking robots. The end-effector of the picking robot developed is a new cable-driven non-contact picking fixture with sensing function, which is composed of three active fingers, three passive covering fingers, and a cutter mechanism. The robot uses a collision-free motion planning algorithm to make picking safer and more convenient. For the picking of the cluster-shaped fruit of litchi, the Zhou team [15] of South China Agricultural University developed a picking robot whose end effector is mainly composed of an end-holder and a rotating cutter head. The robot uses a collision-free motion planning algorithm to make picking safer and more convenient. For the picking of cherry tomato, which is also a cluster fruit, Feng et al. [16] developed a scissor-like end-effector, with two cutters used to cut the stalk. By closing or opening the grip fixed to the cutter, the fruit can be cut and processed reliably.

For tomato picking, Yurni Oktarina et al. [17] in Indonesia designed a tomato picking robot. The end effector is a pair of scissors, sharp and flexible, which is driven by a servo motor. The strawberry-picking robot developed by Xiong's team [14] opens its mold cavity to "swallow" the fruit when picking, and it cuts the fruit stem with a blade to complete strawberry picking. For apple mold cavity-sleeve picking, the team designed a spherical two-finger structure gripper that can effectively reduce the fruit damage rate [18]. The team also developed a flexible gripper composed of two curved flexible fingers, and they improved and optimized it so that it could not only pick apples, but also pick pomegranates, grapefruits, and other fruits [19]. In order to further reduce the damage rate of apple picking, the team studied a bionic three-finger flexible gripper inspired by the octopus tentacle [20]. The end-effector of the pneumatic finger clip structure developed by Hohimer et al. [21] can pick apples flexibly and with high precision. Yu et al. [22] designed three-finger grippers made of flexible materials from an ergonomic point of view, being beneficial to protect apples from damage and achieve non-destructive picking. Due to the large difference in agronomic characteristics between fruits, the use outside the scope of the application scenario will greatly affect the integrity and picking efficiency of the fruit. Therefore, based on the above fruit pickers such as strawberry and tomato, an integrated end-picker for the trellised pear garden scene was developed.

The development of multiple robotic arms provides new ideas for picking research. The cherry-picking robot based on multi-joint robot arms developed by the ArimaS team utilizes a visual system to identify obstacles in the environment and perform path planning for a single robot arm [23,24]. Wageningen University developed a six-degree-of-freedom cucumber-picking robot that uses an identification device on the end effector to identify fruit stems and picks cucumbers by clamping the fruit stems and cutting them at high temperatures [25]. DanSteere developed an apple-picking robot in 2015 that is fast, efficient, and has a wide working range. It uses a four-degree-of-freedom robot arm and an air-suction end-effector [26]. Due to the limitation of picking efficiency of a single robot arm, the advantages of collaborative picking by multiple robotic arms are particularly prominent.

In order to improve the efficiency of apple picking, FFRobotics developed a parallel multiple robot arm apple-picking platform that adopts the way of grouping and dividing the working area [27]. In order to improve the efficiency of apple picking, FFRobotics developed an apple-picking platform based on a parallel multiple robot arm, which adopts the working mode of grouping and work area division [27]. This method can improve the picking efficiency and avoid the interference and collision between robot arms. Williams et al. [28] explored a kiwifruit-picking robot with four three-degree-of-freedom serial robot arms. The robot can collaboratively pick while effectively avoiding collisions between the robot arms. On the other hand, Fu et al. [29] developed a system containing four three-degree-of-freedom rectangular coordinate robots and applied it to the collaborative picking of kiwifruit. For strawberries, a fruit that is difficult to harvest, Harvest CROO

developed a strawberry-harvesting robot for high-ridge cultivation that uses four parallel picking units. Each ridge of strawberries is equipped with an independent picking unit, thus improving work efficiency [30]. AGROBOT Robotics developed a robot suitable for elevated cultivation, using a solution of 24 parallel robot arms. The linear module utilizes mechanical isolation in the forward direction to avoid interference between robot arms, while robot arms on the same unit adopt a control strategy to achieve isolation [31].

As the process of population aging intensifies, a labor-intensive industry like picking urgently needs to be replaced by more efficient multiple robot arm systems. This paper focuses on the collaborative picking of multiple robotic arms in a trellised pear orchard. Taking the trellised pear garden as the object of research, the environment, agronomy, and physical characteristics of the fruit and the identification and positioning method of the pear were investigated. Based on the physical characteristics of pears, a pick-place integrated end-picker was designed. Based on kinematic analysis of robot arms and the construction of a private dataset, the YOLOv5s object detection algorithm was utilized in conjunction with a depth camera to achieve fruit positioning, and the hand-eye system calibration was carried out. It meets the research and development needs of an efficient picking robot in a pear orchard. In order to solve the optimal picking sequence, as well as reduce the scheduling time and the energy consumption, a task allocation method for a multiple robotic arms system was proposed, and the picking sequence was optimized through the simulated annealing algorithm, and finally, the experimental environment was set up for the robot arm picking experiment. The comparison experiment between different end-pickers and the comparison experiment before and after picking sequence optimization were conducted to test the efficiency of different end-pickers. On this basis, the task allocation method and optimal configuration were analyzed and verified by simulation experiments. This research can be utilized to advance robotic pear-picking research and development.

2. Pick-Place Integrated End-Picker

Kinematic analysis of the picking robot arm is the basis for trajectory planning and motion control in picking actions. Picking kinematics analysis mainly includes forward kinematics analysis and inverse kinematics analysis. Kinematics analysis describes the mapping relationship between the robot arm joint coordinate system and Cartesian space. Forward kinematics analysis is based on the rotation angle of each joint of a robot arm combined with the adjacent connecting rod coordinate system transformation matrix to obtain the mapping relationship between the robot arm picking end and base coordinate system. Inverse kinematics analysis is used to calculate the angle of each joint of a robot arm based on the mapping relationship between the picking end of the robot arm and the base coordinate system, and this solution often has several groups. At the same time, for the trellised pear garden, a pick-collect integrated end-picker was designed, the 3D model was established, and the physical processing was completed by 3D printing technology.

2.1. Kinematics of Robot Arm

2.1.1. Kinematic Analysis of the Robot Arm

In this paper, the improved D-H modeling method was utilized in kinematic modeling. The implementation steps are as follows:

To determine the coordinate system position between the connecting rods of a robot arm, the coordinate system was established according to the following criteria:

1. The Z_i axis coincides with the axis of the joint i ; if the Z_i axis and Z_{i+1} axis intersect, the intersection point of the two axes is the origin of coordinate system, and if not, the origin is the intersection of the common perpendicular of the two axes and Z_i axis. The X_i axis is perpendicular to the Z_i axis and Z_{i+1} axis. If the Z_i axis and Z_{i+1} axis do not intersect, X_i axis points from the Z_i axis to the Z_{i+1} axis. After the Z_i axis and X_i axis are determined, the Y_i axis can be determined according to the right-hand rule.

The connecting rod coordinate systems of a robot arm can be determined through the above steps. To simplify the transformation of a coordinate system, the third and fourth coordinate systems are offset (Figure 1).

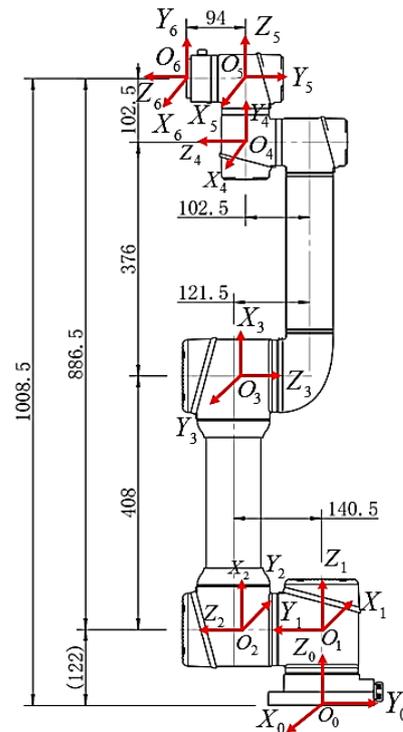


Figure 1. The connecting rod coordinate system.

2. Transforming the coordinate system of two adjacent connecting rods by translation and rotation. The implementation steps in the improved D-H modeling method are as follows: (1) Rotating the coordinate system $(X_{i-1}, Y_{i-1}, Z_{i-1})$ around the X_{i-1} axis so that the Z_{i-1} axis is parallel to the Z_i axis. (2) Translating the coordinate system $(X_{i-1}, Y_{i-1}, Z_{i-1})$ along the direction of the X_{i-1} axis until the Z_{i-1} axis coincides with the Z_i axis. (3) Rotating the coordinate system $(X_{i-1}, Y_{i-1}, Z_{i-1})$ around the Z_i axis so that the X_{i-1} axis is parallel to the X_i axis. (4) Translating the coordinate system $(X_{i-1}, Y_{i-1}, Z_{i-1})$ d_i along the Z_i axis so that the X_{i-1} axis coincides with the X_i axis. After the coordinate systems coincide, we determined the D-H parameters (Table 1).

Table 1. D-H parameter table.

Connecting Rod Number	Torsion Angle (°)	Length of Connecting Rod (m)	Joint Angle (°)	Setover of Connecting Rod (m)
1	0	0	180	0.122
2	−90	0	90	0.1215
3	180	0.408	0	0
4	180	0.376	−90	0
5	−90	0	0	0.1025
6	90	0	0	0.094

The calculating formula for the transformation matrix between the adjacent connecting rod coordinate systems is expressed in Equation (1):

$$\begin{aligned}
 {}^{i-1}T_i &= R(X_{i-1}, \alpha_{i-1})T(X_{i-1}, \alpha_{i-1})R(Z_i, \theta_i)T(Z_i, d_i) \\
 &= \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & a_{i-1} \\ \cos(\alpha_{i-1}) \sin(\theta_i) & \cos(\alpha_{i-1}) \cos(\theta_i) & -\sin(\alpha_{i-1}) & -d_i \sin(\alpha_{i-1}) \\ \sin(\alpha_{i-1}) \sin(\theta_i) & \cos(\theta_i) \sin(\alpha_{i-1}) & \cos(\alpha_{i-1}) & d_i \cos(\alpha_{i-1}) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)
 \end{aligned}$$

3. Obtaining the transformation matrix ${}^0_1T \quad {}^1_2T \quad {}^2_3T \quad {}^3_4T \quad {}^4_5T \quad {}^5_6T$ between the connecting rods from the D-H parameter table.
4. Obtaining the transformation matrix between any two connecting rods by continuous multiplication from the transformation matrix between the adjacent connecting rods.

$${}^n_mT = {}^n_{n+1}T {}^{n+1}_{n+2}T {}^{n+2}_{n+3}T \dots {}^{m-1}_mT, \text{ where } n < m.$$

2.1.2. Analysis of forward and Inverse Kinematics

The forward kinematics of a robot arm can calculate the position and attitude of the end-picker relative to the base coordinate system through the known angle information of each joint. The process is obtained by successive multiplication the transformation matrices of adjacent connecting rod coordinate systems:

$${}^0_6T = {}^0_1T {}^1_2T {}^2_3T {}^3_4T {}^4_5T {}^5_6T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

The inverse kinematics of a robot arm means obtaining the rotation angle of each robot arm joint based on the given transformation matrix of the end-picker relative to the base coordinate system. That is to say, on the premise of known 0_6T , seeking the solution of angles $\theta_1 - \theta_2$. In order to solve the six angles, we multiply the above equations on the left by ${}^0_1T^{-1}$:

$${}^1_6T = {}^0_1T^{-1} {}^0_6T = \begin{bmatrix} \cos(\theta_1) & \sin(\theta_1) & 0 & 0 \\ -\sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & -d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} R_{11}' & R_{12}' & R_{13}' & P_x' \\ R_{21}' & R_{22}' & R_{23}' & P_y' \\ R_{31}' & R_{32}' & R_{33}' & P_z' \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

Among them, each parameter can be calculated as

$$\begin{aligned} R_{11}' &= r_{11} \cos(\theta_1) - r_{21} \sin(\theta_1) & R_{13}' &= r_{13} \cos(\theta_1) + r_{23} \sin(\theta_1) \\ R_{21}' &= r_{21} \cos(\theta_1) - r_{11} \sin(\theta_1) & R_{23}' &= r_{23} \cos(\theta_1) - r_{13} \sin(\theta_1) \\ R_{31}' &= r_{31} & R_{33}' &= r_{33} \\ R_{12}' &= r_{12} \cos(\theta_1) + r_{22} \sin(\theta_1) & P_x' &= p_x \cos(\theta_1) + p_y \sin(\theta_1) \\ R_{22}' &= r_{22} \cos(\theta_1) - r_{12} \sin(\theta_1) & P_y' &= p_y \cos(\theta_1) - p_x \sin(\theta_1) \\ R_{32}' &= r_{32} & P_z' &= p_z - d_1 \end{aligned} \tag{4}$$

In addition, it can be obtained by direct successive multiplication of the transformation matrices of the adjacent connecting rod coordinate systems:

$${}^1_6T = {}^1_2T {}^2_3T {}^3_4T {}^4_5T {}^5_6T = \begin{bmatrix} R_{11} & R_{12} & R_{13} & P_x \\ R_{21} & R_{22} & R_{23} & P_y \\ R_{31} & R_{32} & R_{33} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5}$$

According to the principle that corresponding elements of the matrix are equal, six angle values can be calculated:

Solving θ_1 :

Let matrix elements R_{23}' and R_{23} be equal, and elements R_{24}' and R_{24} be equal, as follows:

$$\begin{aligned} r_{23} \cos(\theta_1) - r_{13} \sin(\theta_1) &= \cos(\theta_5) \\ p_y \cos(\theta_1) - p_x \sin(\theta_1) &= d_2 + d_6 \cos(\theta_5) \end{aligned} \tag{6}$$

The above results imply that

$$(p_y - d_6 r_{23}) \times \cos(\theta_1) + (d_6 r_{13} - p_x) \times \sin(\theta_1) = d_2 \tag{7}$$

Let $p_y - d_6 r_{23} = r \sin(\varphi) = B$, $d_6 r_{13} - p_x = r \cos(\varphi) = A$, $r = \sqrt{A^2 + B^2}$, and then we have from a triangular replacement:

$$\begin{aligned} \sin(\theta_1 + \varphi) &= d_2/r \\ \cos(\theta_1 + \varphi) &= \pm\sqrt{1 - d^2/r^2} \end{aligned} \tag{8}$$

When $1 - d^2/r^2 \geq 0$, θ_1 has a solution, which is

$$\theta_1 = a \tan 2(d_2, \pm\sqrt{r^2 + d_2^2}) - a \tan 2(B, A) \tag{9}$$

Solving θ_5 :

After θ_1 is calculated, we obtain the following according to Formula (6):

$$\begin{aligned} \sin(\theta_5) &= \pm\sqrt{1 - (r_{23} \cos(\theta_1) - r_{13} \sin(\theta_1))^2} \\ \theta_5 &= a \tan 2\left(\pm\sqrt{1 - (r_{23} \cos(\theta_1) - r_{13} \sin(\theta_1))^2}, r_{23} \cos(\theta_1) - r_{13} \sin(\theta_1)\right) \end{aligned} \tag{10}$$

Solving angle θ_6 :

Let matrix elements R_{21}' and R_{21} be equal, and elements R_{22}' and R_{22} be equal, as follows:

$$\begin{aligned} r_{21} \cos(\theta_1) - r_{11} \sin(\theta_1) &= -\sin(\theta_5) \times \cos(\theta_6) \\ r_{22} \cos(\theta_1) - r_{12} \sin(\theta_1) &= \sin(\theta_5) \times \sin(\theta_6) \end{aligned} \tag{11}$$

The above results imply that

$$\begin{aligned} \cos(\theta_6) &= \frac{r_{11} \sin(\theta_1) - r_{21} \cos(\theta_1)}{\sin(\theta_5)} \\ \sin(\theta_6) &= \frac{r_{22} \cos(\theta_1) - r_{12} \sin(\theta_1)}{\sin(\theta_5)} \end{aligned} \tag{12}$$

When $\sin(\theta_5) \neq 0$, θ_6 has a solution, which is

$$\theta_6 = a \tan 2\left(\frac{r_{22} \cos(\theta_1) - r_{12} \sin(\theta_1)}{\sin(\theta_5)}, \frac{r_{11} \sin(\theta_1) - r_{21} \cos(\theta_1)}{\sin(\theta_5)}\right) \tag{13}$$

Solving θ_2 :

Let matrix elements R_{13}' and R_{13} be equal, and elements R_{33}' and R_{33} be equal, as follows:

$$\begin{aligned} r_{13} \cos(\theta_1) + r_{23} \sin(\theta_1) &= \cos(\theta_2 - \theta_3 + \theta_4) \times \sin(\theta_5) \\ r_{33} &= -\sin(\theta_2 - \theta_3 + \theta_4) \times \sin(\theta_5) \end{aligned} \tag{14}$$

The above results imply that

$$\begin{aligned} \cos(\theta_2 - \theta_3 + \theta_4) &= \frac{r_{13} \cos(\theta_1) + r_{23} \sin(\theta_1)}{\sin(\theta_5)} \\ \sin(\theta_2 - \theta_3 + \theta_4) &= \frac{-r_{33}}{\sin(\theta_5)} \\ \theta_2 - \theta_3 + \theta_4 &= a \tan 2\left(\frac{-r_{33}}{\sin(\theta_5)}, \frac{r_{13} \cos(\theta_1) + r_{23} \sin(\theta_1)}{\sin(\theta_5)}\right) \end{aligned} \tag{15}$$

Let matrix elements P_x' and P_x be equal, and elements P_z' and P_z be equal, as follows:

$$\begin{aligned} p_x \cos(\theta_1) + p_y \sin(\theta_1) &= a_2 \cos(\theta_2) - d_5 \sin(\theta_2 - \theta_3 + \theta_4) + a_3 \cos(\theta_2 - \theta_3) + d_6 \sin(\theta_5) \cos(\theta_2 - \theta_3 + \theta_4) \\ p_z - d_1 &= -a_2 \sin(\theta_2) - d_5 \cos(\theta_2 - \theta_3 + \theta_4) - a_3 \sin(\theta_2 - \theta_3) - d_6 \sin(\theta_5) \sin(\theta_2 - \theta_3 + \theta_4) \end{aligned} \tag{16}$$

By transposition, it follows that

$$\begin{aligned} a_2 \cos(\theta_2) + a_3 \sin(\theta_2 - \theta_3) &= p_x \cos(\theta_1) + p_y \sin(\theta_1) + d_5 \sin(\theta_2 - \theta_3 + \theta_4) - d_6 \sin(\theta_5) \cos(\theta_2 - \theta_3 + \theta_4) \\ a_2 \sin(\theta_2) + a_3 \sin(\theta_2 - \theta_3) &= d_1 - d_6 \sin(\theta_5) \sin(\theta_2 - \theta_3 + \theta_4) - d_5 \cos(\theta_2 - \theta_3 + \theta_4) - p_z \end{aligned} \tag{17}$$

Given

$$p_x \cos(\theta_1) + p_y \sin(\theta_1) + d_5 \sin(\theta_2 - \theta_3 + \theta_4) - d_6 \sin(\theta_5) \cos(\theta_2 - \theta_3 + \theta_4) = C$$

$$d_1 - d_6 \sin(\theta_5) \sin(\theta_2 - \theta_3 + \theta_4) - d_5 \cos(\theta_2 - \theta_3 + \theta_4) - p_z = D \quad (18)$$

we obtain

$$C \cos(\theta_2) + D \sin(\theta_2) = \frac{C^2 + D^2 + a_2^2 - a_3^2}{2a_2} = E \quad (19)$$

Let

$$\sin \beta = \frac{C}{\sqrt{C^2 + D^2}}, \cos \beta = \frac{D}{\sqrt{C^2 + D^2}}, \quad (20)$$

and then we have from triangular replacement the following:

$$\sin(\beta + \theta_2) = \frac{E}{\sqrt{C^2 + D^2}}$$

$$\cos(\beta + \theta_2) = \pm \sqrt{1 - E^2 / (C^2 + D^2)} \quad (21)$$

$$\theta_2 = a \tan 2(E, \pm \sqrt{C^2 + D^2 - E^2}) - a \tan 2(C, D)$$

When $C^2 + D^2 - E^2 \geq 0$, θ_2 has a solution.

Compute θ_3 :

After solving θ_3 , compute $\theta_2 - \theta_3$:

$$\theta_2 - \theta_3 = a \tan 2\left(\frac{D - a_2 \sin(\theta_2)}{a_3}, \frac{C - a_2 \cos(\theta_2)}{a_3}\right)$$

$$\theta_3 = \theta_2 - (\theta_2 - \theta_3) \quad (22)$$

Solving θ_4 :

$$\theta_4 = (\theta_2 - \theta_3 + \theta_4) - (\theta_2 - \theta_3) \quad (23)$$

All joint angles have been solved. A total of eight groups of inverse solutions can be solved, and a group of the most suitable solutions can be selected according to the size of the angle spinor.

2.2. The Design of the Executive Mechanism

In this study, the mold cavity-sleeve structure was used for pear picking. The active cutting mode will greatly increase the structural complexity of the end actuator, and the mold cavity sleeve itself will occupy a larger volume. If the cutting device is added, the volume and weight of the actuator will be increased. Therefore, a passive cutting mode for which the end actuator cooperates with the robot arm to complete the picking action was adopted to realize the miniaturization of the end-picker. By rational design of the opening and closing mechanism above the end-picker and cooperating with the robot arm, the crown pear stems can be cut. Figure 2 shows the overall assembly diagram of the end-picker.

Figure 3a shows the three-dimensional structure of the blade. The cutting mechanism of the end-picker designed for mold cavity picking is the key to separating pears from stems. There are two connecting holes on both sides of the structure that are connected to main body of the mold cavity. Two blades are arranged symmetrically, wherein opening and closing are controlled by the steering engine. After the fruit enters the mold cavity, the steering engine drives blades to clamp the stem and reset after picking is completed.

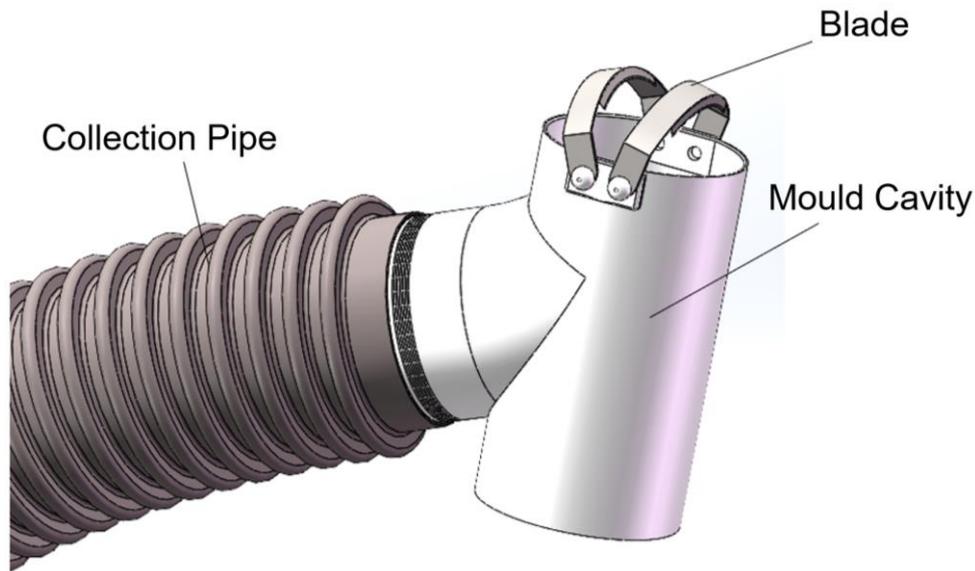


Figure 2. End-picker assembly diagram.

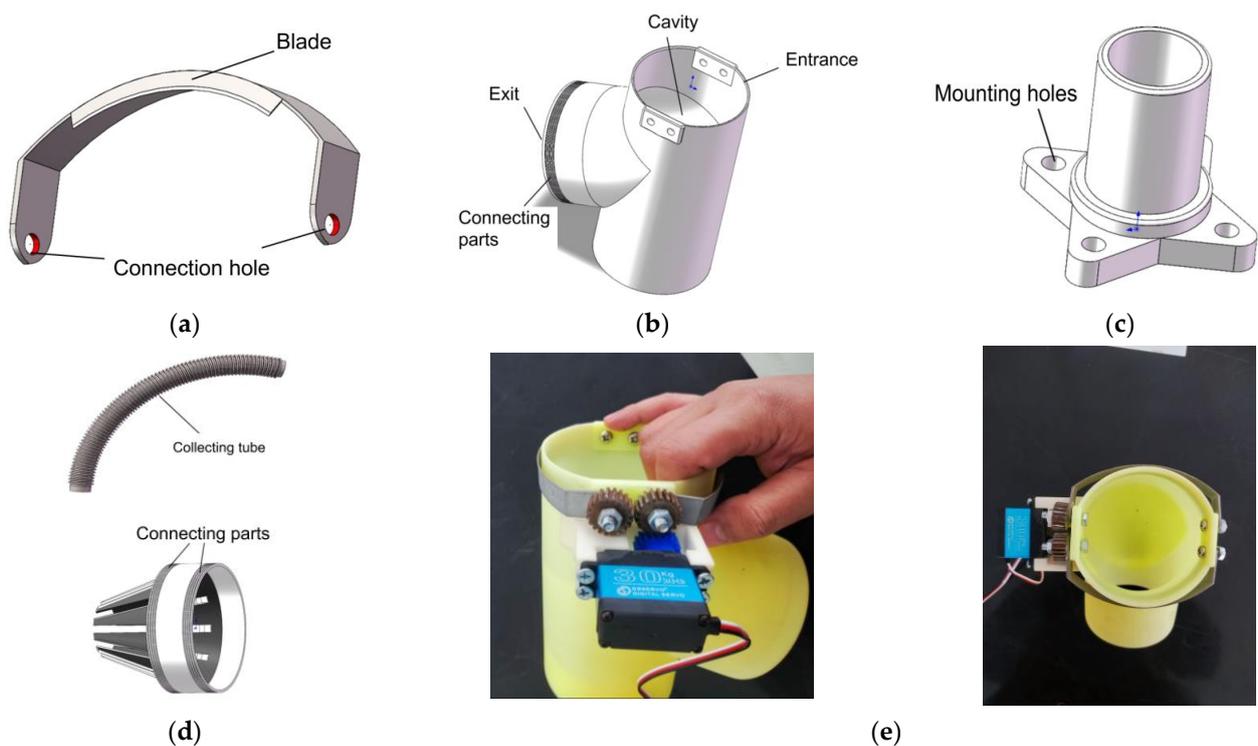


Figure 3. End-picker assembly diagram: (a) blade; (b) mold cavity; (c) connecting mechanism; (d) fruit-collecting mechanism; (e) transmission mechanism and assembly figure.

Figure 3b shows the mold cavity of the picker. The main body is cylindrical. The fruit entrance and exit of the mold cavity is a connecting tube with an inner diameter of 100 mm. After fruit enters from the inlet, it will directly enter the fruit-collecting tube from the outlet. As long as the fruit enters the mold cavity, there will be no damage to the fruit due to the robot arm not grasping it firmly. The bottom of the cavity of the picker is closed and exists as a buffer structure in the internal structure. A slope channel is formed in the cavity with foam and other buffering materials so that the fruit can be buffered and enter the collection tube along the ramp. Therefore, the mold cavity-sleeve method can protect the fruit well and avoid damage to the fruit during the picking process.

Figure 3c shows the connection mechanism, which is used to connect the end-picker and the end flange of the robot arm. The four mounting holes correspond to the four threaded holes of the end flange of the robot arm.

Figure 3d shows the collecting mechanism. The diameter of the collecting tube is 100 mm. And the tube is made of rubber to prevent pears from being damaged by friction with the tube inner wall during the collecting process. The deceleration mechanism is set in the middle of the collecting tube. When the fruit passes through, its kinetic energy will be reduced, thereby reducing the speed in the tube and the loss of fruit.

Figure 3e shows the transmission mechanism and photos illustrating the installation of each element. The transmission mechanism mainly has three gears, for which the module is 1, the number of teeth is 20, and the inner diameter of the hole is 6 mm. Two metal gears are fixed at the two blade mounting holes and mesh with each other. The driving mode of the steering gear is PWM control, which can accurately control the predetermined rotation angle. It is mainly used to drive the meshing gear fixed on the blade and then drive the opening and closing of the two blades for picking.

3. Target Detection and 3D Location

3.1. Dataset Construction

At present, there are few open-source pear datasets in the field of agricultural picking, and especially the data of the natural unpicked state and fruit under a trellised orchard are very scarce. The experimental dataset was collected by the orchard and the trellised pear orchard environment built in the laboratory. The image format was JPEG with a resolution of 3024×4032 , and the video format was MP4 with a resolution of 4K. And the diversity of the dataset was improved by collecting datasets in two scenes of natural light during the day and filling light at night.

The visual equipment used for target detection and calibration in this paper was an Intel RealSense D435i depth camera, which was composed of an rgb camera, infrared camera, infrared transmitter, and IMU (depth resolution: 1280×720 ; RGB sensor resolution: 1920×1080).

The performance of deep learning models strongly depends on the quantity and quality of input data. Many deep learning projects fail to obtain enough data in practical applications, resulting in the underfitting of the model. In order to improve the model generalization ability, that is, the ability to predict new data, data augmentation techniques are widely used. Currently, supervised data augmentation schemes are commonly used, which operate the dataset through known image transformation rules to make the model learn more features and increase the richness of the dataset, thereby improving the performance and stability of the model. The data augmentation methods used in this article include (1) random cropping; (2) mirror image; (3) Gaussian noise; (4) grayscale image; and (5) rotating 90° .

Each image collected was manually screened and classifiatorily annotated by a label using Labellmg software (v1.8.1) as shown in Figure 4. In the annotation process, a rectangle annotation method is used to draw the minimum bounding rectangle of the pear object in the image, and the categorical attribute of the rectangle frame is set to "pear". After the annotation is completed, the generated xml format label file is automatically saved, and the xml format text contains the height, width, and categorical information of the rectangular box. According to the VOC2007 dataset format, the trellised pear orchard dataset was made, wherein the dataset has a total of 2574 images, and the ratio of the training set to validation set was set to 7:3.



Figure 4. Dataset annotation.

3.2. Yolo-v5s Detection Model

Experimental running environment: CPU: i7-12700H, 16GB running memory; GPU: GTX 3060, 6 GB graphic memory; running system: Windows11; depth framework: PyTorch, GPU1.10.1, with CUDA11.6 and cudnn7.6.5. Running software: Pycharm2021.3 community edition, Python3.8.16.

Considering the experimental hardware platform conditions and model detection accuracy, the choice of network depth and width is very important. Under the condition of the minimum depth and width of the model, the YOLOv5s model is a very excellent object detection model that can achieve high-precision object detection with a smaller memory footprint and computing resources. The input layer of the YOLOv5s model is used to receive the original image data, and the Backbone layer can extract feature information from the original image, with the Neck layer being able to further compress the feature information. Finally, object detection and classification are performed in the Detect layer. The structure of the YOLOv5s model used in this project is shown in Figure 5.

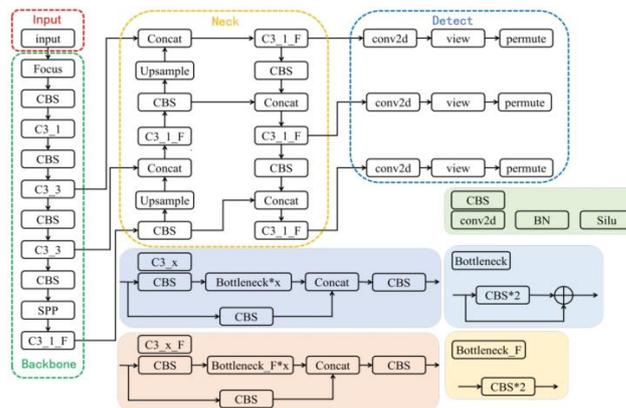


Figure 5. The YOLOv5s model structure.

Based on the collected dataset above and the model, the results of network training by setting parameters are shown in the figure. Figure 6a shows the confusion matrix, which is a commonly used method to evaluate the performance of classification models and is usually used to calculate precision, recall, F1 score, and other indicators of classification algorithms. The horizontal axis of the confusion matrix represents the real label, the vertical axis represents the predicted label, and the value of each cell represents the number of samples that the real label matches to the predicted label. The probability of the pears in Figure 6a being correctly classified is 0.99. In Figure 6b, there is shown a certain relationship between F1 score and confidence thresholds. F1 score can help to comprehensively consider the performance of the model in the classification task, with neither too much emphasis

on precision, nor too much emphasis on recall, so that the detection results of models are more balanced and reliable. Typically, the top of the curve is close to 1, indicating that the model performs well on the training dataset. Figure 6c shows the precision–confidence threshold curve. The higher confidence is, the higher the precision of the classification detection is. But, it is possible to miss some real samples with low probability of judgment. Figure 6d shows the precision–recall curve (PR curve), which is also a commonly used performance evaluation tool for evaluating the classification performance of the model in multiple categories. As can be seen from the figure, the higher precision is, the lower recall is. It is hoped that all categories can be detected as far as possible under the premise of high precision. Therefore, the area of the mAP curve should be as close to 1 as possible.

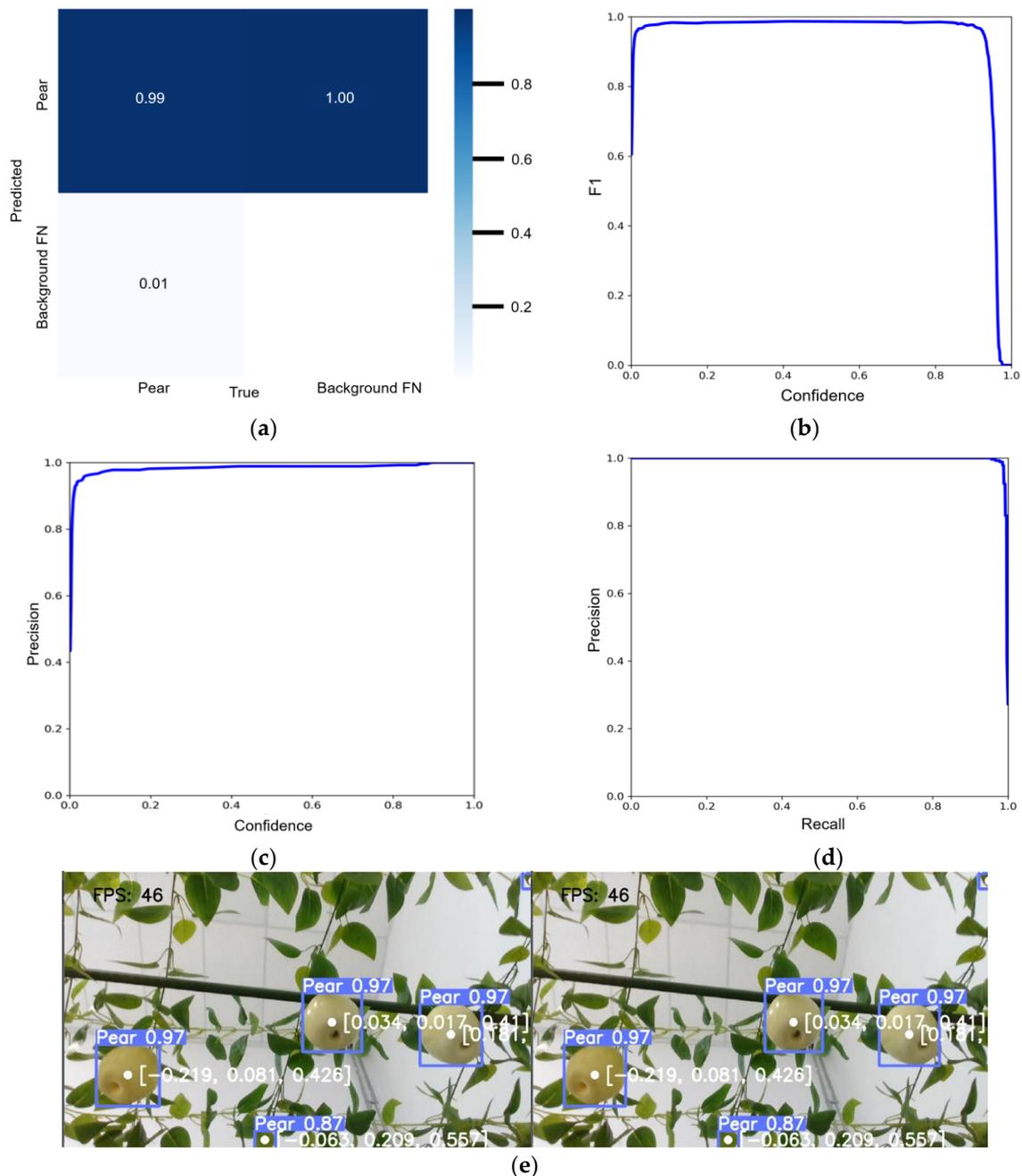


Figure 6. Performance evaluation. (a) Confusion matrix; (b) F1–confidence curve; (c) precision–confidence curve; (d) PR curve; (e) object detect results.

3.3. Hand–Eye System Calibration

In the process of camera calibration and hand–eye calibration, the transformation relationship of four coordinate systems is involved, which are the pixel coordinate system, image coordinate system, camera coordinate system, and world coordinate system, as shown in Figure 7. For this study, the world coordinate system was the robot arm base coordinate system.

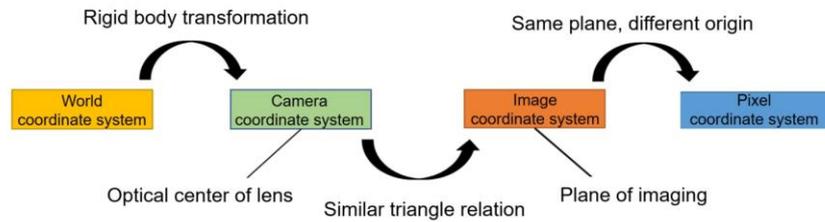


Figure 7. Coordinate transformation.

Hand–eye calibration mainly finds the relationship between the camera coordinate system and the robot arm base coordinate system. By deriving the transformation matrix between these two coordinate systems, the 3D coordinates of the detected target in the camera coordinate system can be converted to the robot arm base coordinate system. There are mainly two hand–eye calibration methods, as shown in Figure 8, which can be divided into “eyes in hand” and “eye to hand”, depending on the setting position of the camera.

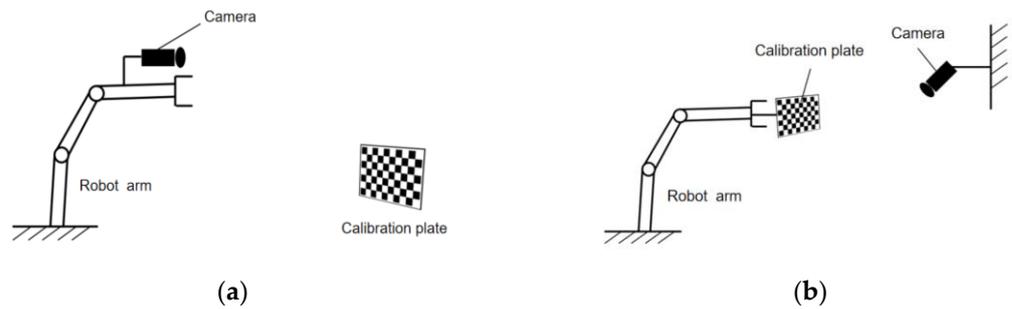


Figure 8. Setting position of the camera. (a) Eyes in hand; (b) eye to hand.

For the “eyes to hand” method of hand–eye calibration, during the movement of the robot arm, the camera coordinate system and the robot arm base coordinate system are fixed, and the transformation matrix between them is also unchanged. Hand–eye calibration is used to solve the transformation relationship between these two coordinate systems, as shown in Figure 9.

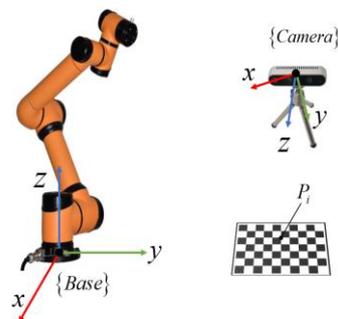


Figure 9. Calibration diagram.

We set the robot arm base coordinate system to {Base} and the camera coordinate system to {Camera}. If the coordinates of several fixed points P_i under these two coordinate

systems are known, the corresponding transformation matrix of these two coordinate systems can be derived according to the coordinate transformation formula. The implementation steps are as follows:

- (1) We set the coordinate of a spatial point P_i under {Base} to $P_i(x'_i, y'_i, z'_i)$ and the coordinate under {Camera} to $P_i(x''_i, y''_i, z''_i)$. On the basis of the relationship between these two coordinates and the homogeneous matrix form, the following formula can be derived:

$$\begin{cases} r_{11}x_i + r_{12}y_i + r_{13}z_i + a = x'_i \\ r_{21}x_i + r_{22}y_i + r_{23}z_i + b = y'_i \\ r_{31}x_i + r_{32}y_i + r_{33}z_i + c = z'_i \\ 1 = 1 \end{cases} \quad (24)$$

- (2) We derive the pixel coordinates of the spatial point P_i by the object detection algorithm, and the specific coordinates of the point in the camera coordinate system by combining the camera internal parameters and the depth information of the depth camera. The coordinates in the robot arm base coordinate system could be read out directly by the robot arm teach pendant.
- (3) We select twenty different points to fit the optimal results, solving the following parameters: $r_{11}, r_{12}, r_{13}, r_{21}, r_{22}, r_{23}, r_{31}, r_{32}, r_{33}, a, b, c$, as well as listing the overdetermined linear equations of these points in the camera coordinate system.
- (4) We solve the overdetermined equations by the least squares method. We obtain the rotation matrix and translation vector between the camera coordinate system and the robot arm base coordinate system.

4. The Picking Task Planning of Multiple Robot Arms

4.1. The Task Allocation Method of Multiple Robot Arms

In this paper, the Monte Carlo method based on random probability was utilized to solve the working space of a robot arm. Figure 10 shows the working space of a single AUBO-i5 robot arm and the projection of the working space on xoz. The working space is approximately a sphere, and the picking area is divided based on this.

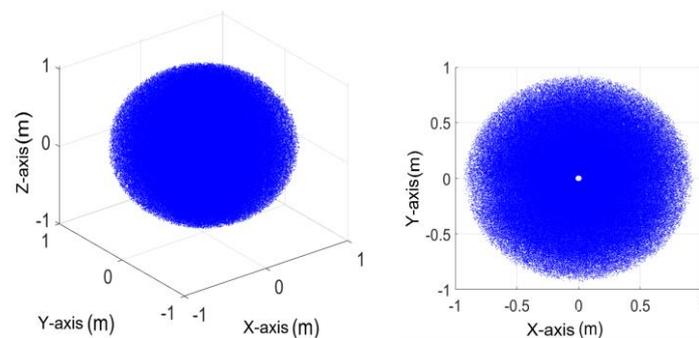


Figure 10. The working space of AUBO-i5.

It can be seen from Figure 11 that when the picking plane was 0.4009 m away from the unmanned vehicle plane, the radius of the picking plane was 0.8703 m; when the picking plane was 0.923 m away from the unmanned vehicle plane, the radius of the picking plane was 0.4696 m. As picking height increases, the picking range will decrease. The projection of the workspace on xoy, the circular area, decreased as the Z-axis absolute value increased.

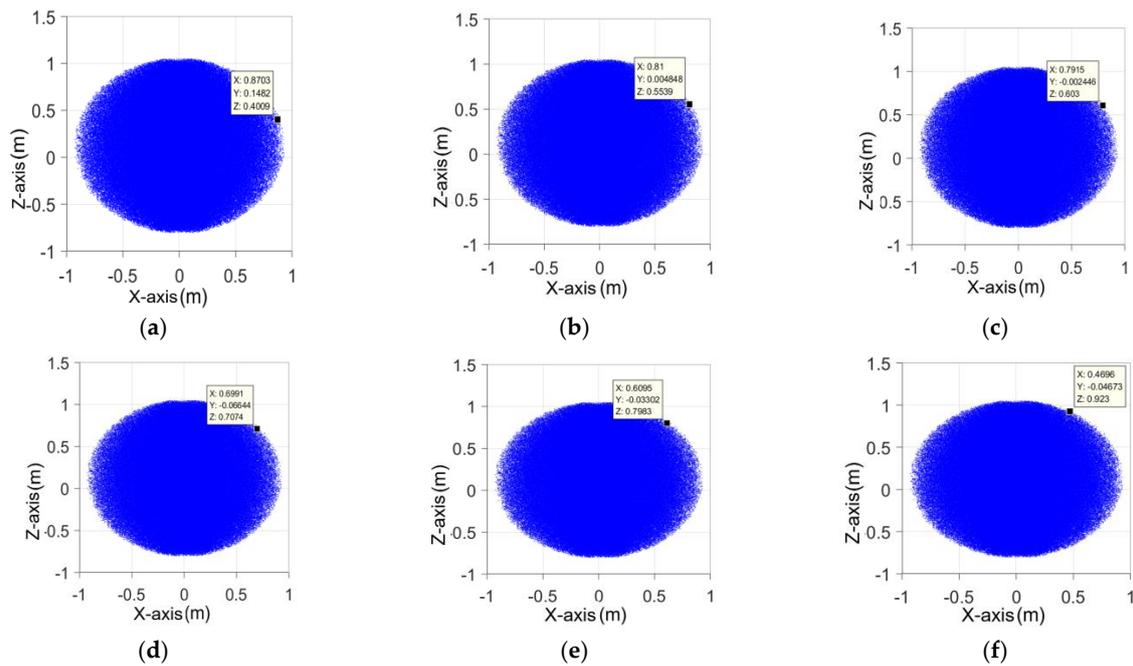


Figure 11. The analysis diagram of a picking area with a single robot arm. (a) $z = 0.4$ m, $x = 0.87$ m; (b) $z = 0.55$ m, $x = 0.81$ m; (c) $z = 0.60$ m, $x = 0.80$ m; (d) $z = 0.71$ m, $x = 0.70$ m; (e) $z = 0.80$ m, $x = 0.61$ m; (f) $z = 0.92$ m, $x = 0.47$ m.

The distance between robot arms was set to 1 m. The projection of a multiple robot arm workspace on xoy can be obtained using the Monte Carlo method (Figure 12).

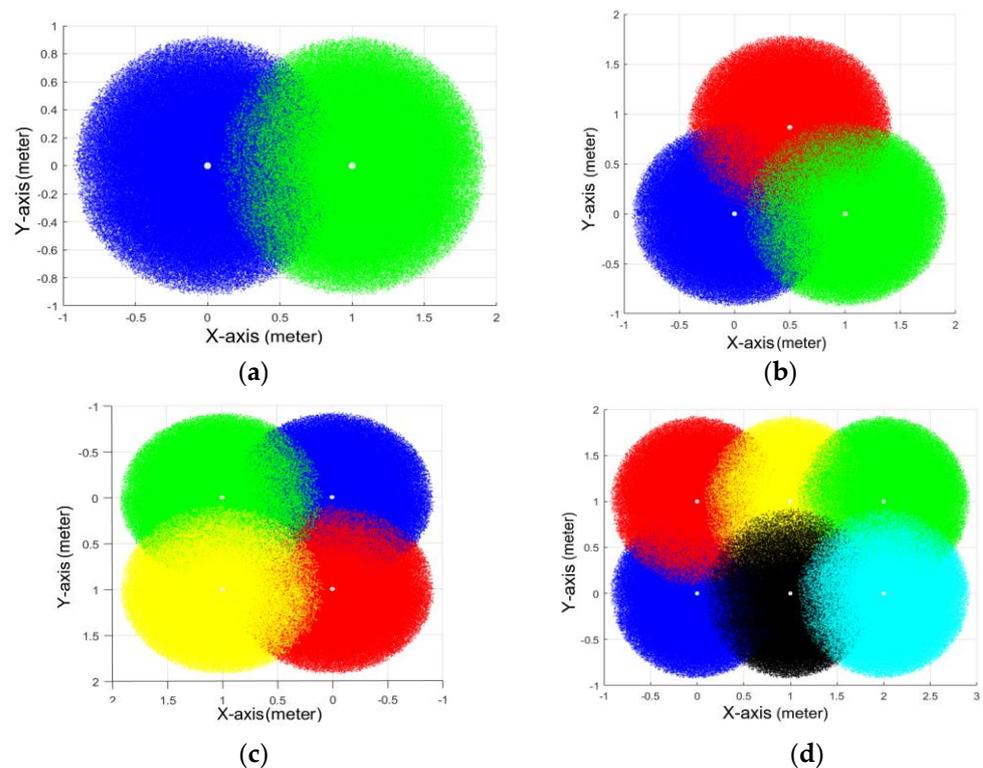


Figure 12. The analysis diagram of a picking range with multiple robot arms (Different colors represent different work areas). (a) the projection of a two-robot-arm workspace; (b) the projection of a three-robot-arm workspace; (c) the projection of a four-robot-arm workspace; (d) the projection of a six-robot-arm workspace.

It can be seen from Figure 12 that for the trellised pear orchard environment, at different heights from the unmanned vehicle plane, the picking area of each robot arm was approximately a circle. As picking height increased, the overlap of the picking area between adjacent robot arms was approximately an ellipse, wherein the area decreased as picking height increased.

The height of an unmanned vehicle is 1.1 m. The robot arm performs picking operations on a picking plane 0.6 m away from the unmanned vehicle plane. The operating area of the robotic arm is approximately a circle with a radius of 0.8 m. On the basis of the solution of multiple robot arm workspace in the previous section, the picking area of a robot arm was divided (Figure 13). Collaborative picking task allocation of multiple robotic arms can be described as the problem of multiple robotic arms cooperating to complete the picking task. The ultimate goal is to increase picking efficiency, and at the same time, tending to divide the task volume of each robot arm evenly and improve the utilization rate of each robot arm.

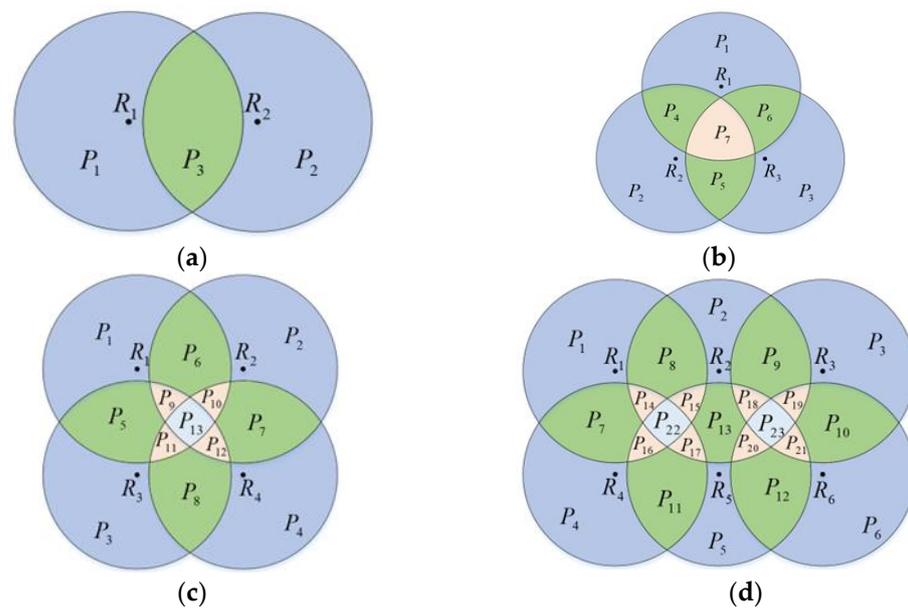


Figure 13. The diagram of a picking area with multiple robot arms. (a) The picking area of a two-robot-arm workspace; (b) the picking area of a three-robot-arm workspace; (c) the picking area of a four-robot-arm workspace; (d) the picking area of a six-robot-arm workspace.

The collaborative picking task allocation model of multiple robotic arms is specifically described as follows:

(1) A set of multiple robotic arms is represented by $R = \{R_1, R_2, R_3, R_4, \dots, R_a\}$, where a represents the number of robot arms and R_i represents the robot arm, $1 \leq i \leq a$. (2) The picking area is represented by $P = \{P_1, P_2, P_3, P_4, \dots, P_b\}$, where b represents the number of picking areas and P_j represents the picking sub-regions, $1 \leq j \leq b$. (3) The task volume in each sub-regions is represented by $G = \{G_1, G_2, G_3, G_4, \dots, G_b\}$, where G_k represents the picking task volume in a sub-picking area k , $1 \leq k \leq b$. (4) Each robot arm is set in different positions, and the picking sub-regions are also different. The set of tasks that the robot arm can pick is $T_{Ri} = \{T_{R1}, T_{R2}, T_{R3}, T_{R4}, \dots, T_{Ra}\}$, where i represents the i -th robot arm, $T_{Ri} \in G$. The task process is shown in Figure 14.

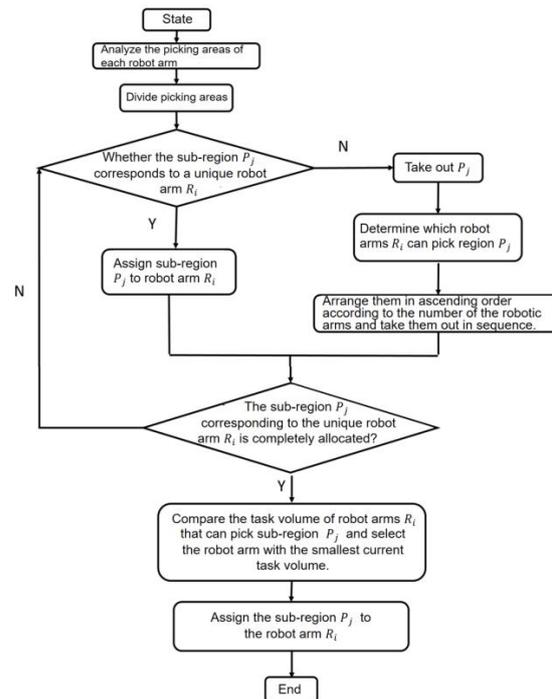


Figure 14. Task allocation flow chart.

Firstly, the number of robot arms, the spacing between the robot arms, and the height of the picking plane are determined. Then, the Monte Carlo method is utilized to analyze the workspace of multiple robot arms and divide the picking area.

After the picking area is divided, the picking area that can only be picked by the only robot arm is first matched with its corresponding one. Then, for other areas that can be picked by two or more robot arms, first, we arrange them from small to large according to the number of robot arms that can pick in this area. Then, we take them out in the order from small to large and select the robot arm with the least task volume until all the areas are allocated. This allocation strategy is mainly to distribute tasks evenly while avoiding collision interference between multiple robotic arms. The purpose of the picking robot arms in this area being taken out in order from small to large is to better balance the task volume of robot arms. This is because as the allocation process gradually proceeds, the number of robot arms corresponding to the picking area increases, and the algorithm can further balance the robot arms with a large difference of task volume assigned at the last time. At the same time, the picking area becomes smaller, and the probability of few fruits in the area becomes greater. So, it will not cause the task volume of robot arm with the fewest tasks to become much higher than that of other robot arms after the allocation. Taking two robot arms as an example, due to uneven fruit distribution in most cases, when the difference of the fruit number between P_1 and P_2 is large, according to the task allocation strategy, the public picking area P_3 will be assigned to the robot arm with a smaller task volume, so that task volume of the two robot arms tends to be evenly allocated, with little difference. The same method will be utilized in the common picking area of three, four, and six robot arms.

4.2. Picking Sequence Optimization

In this paper, in a trellised pear orchard environment, the heights of pears were basically the same. The Z-axis coordinate was ignored, and the multi-fruit picking path optimization problem was converted into a two-dimensional traveling salesman problem. The coordinates of each pear can be given by the depth camera, and then the distance between each two pears can be obtained. The picking path optimization problem is to find

the shortest path that can traverse all the fruits and does not repeat [32]. The TSP model of picking path optimization can be expressed by a mathematical model.

The requirement of the picker design is to complete the most picking in the shortest path. If the simple algorithm is used to pick across the region, the picking path will have great redundancy, and it is more time-consuming to schedule the manipulator back and forth. Therefore, we used the simulated annealing algorithm for optimization, to solve the optimal picking sequence, to reduce the scheduling time of the robot arm, and to reduce energy consumption at the same time.

We set the weighted graph as $G = (V, E)$, and all pear coordinate-sets as $V = \{1, 2, 3, \dots, n - 1, n\}$. E is a set of path weights or lengths, and D_{ij} represents the distance between each pear, where $D_{ij} > 0, i, j \in V$.

$$x_{ij} = \begin{cases} 1, & (i, j) \in L \\ 0, & (i, j) \notin L \end{cases} \tag{25}$$

L is the solution sequence. The TSP mathematical model can be expressed in the following form:

$$\begin{aligned} \text{mins} &= \sum_{i=1}^n \sum_{j=1}^n d_{ij}x_{ij} \\ \text{s.t.} &\begin{cases} \sum_{i=1}^n x_{ij} = 1, j \in V \\ \sum_{j=1}^n x_{ij} = 1, i \in V \\ \sum_{i,j \in S} x_{ij} \leq |S| - 1, \forall S \subseteq V, 1 < |S| < n \end{cases} \end{aligned} \tag{26}$$

The distance between adjacent fruits is set as d_{ij} . S is a non-empty subset of the vertex set V , and $|S|$ is the number of vertices of the set S in the weighted graph G .

This project utilized a simulated annealing algorithm that can avoid falling into local optimality to a certain extent, in order to optimize the picking sequence. The algorithm mainly consists of two parts: one is the Metropolis algorithm, and the other is the annealing cooling process, which corresponds to the internal cycle and the external cycle, respectively.

The algorithm steps are as follows (Figure 15):

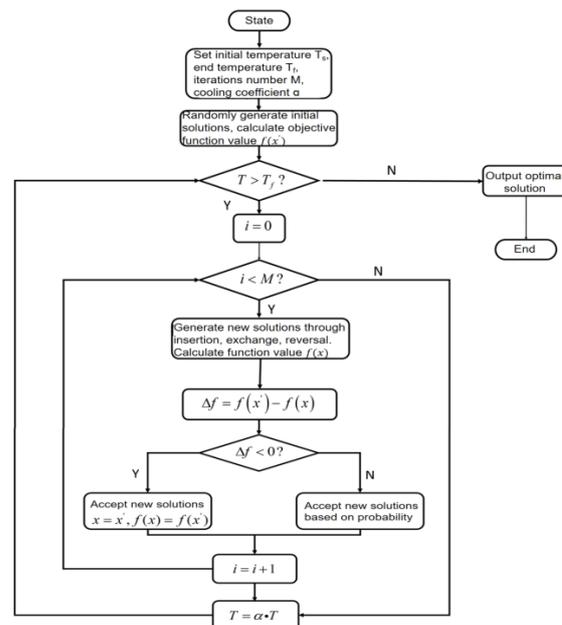


Figure 15. Simulated annealing algorithm flow chart.

(1) Initialize the temperature, and set the current temperature $T = T_s$, the termination temperature T_f , and the maximum number of iterations M . Randomly generate an initial solution x and then calculate the objective function value $f(x) = \sum_{i=1}^n \sum_{j=1}^n d_{ij}$. (2) At a certain temperature, the current solution is perturbed by insertion, exchange, reversal, and other ways to generate a new solution. (3) Solve the objective function value $f(x')$, and calculate $\Delta f = f(x') - f(x)$; if $\Delta f < 0$, then the new solution x' is accepted. If not, the new solution x' is accepted according to the probability. (4) At a certain temperature T , the perturbation and acceptance process is repeated for a certain number of times, that is, steps 2 and 3 are repeated. (5) Lower the temperature T . (6) Repeat steps 2–5 until convergence conditions are met.

The simulated annealing algorithm has three important parameters: initial temperature T_s , internal loop iteration number M , and cooling coefficient α , which determine the optimization capacity and comprehensive performance of the simulated annealing algorithm, that is, finding a better solution quickly and accurately with less time cost. (The above initial temperature does not represent the physical temperature in the real environment, but instead a parameter that needs to be optimized in the simulated annealing algorithm, which represents the base temperature in the optimization algorithm.)

In theory, the larger the initial temperature, the better. A low temperature causes the problem of the algorithm not searching enough in the solution space, resulting in missing the optimal solution in the optimizing process. But if the temperature is too high, it will take a lot of time. The number of inner loop iterations also affects the search ability of the algorithm in the solution space. The greater the number of iterations at the same temperature, the greater the chance of finding the optimal solution, but the time cost is often greatly increased. Then, a too large cooling coefficient easily leads to the loss of the optimal solution, while a too small cooling coefficient greatly increases the algorithm's time cost. The termination temperature determines when the algorithm ends. Setting the termination temperature too high will result in an insufficient search, so the termination temperature is generally set to a smaller value.

(1) The adjustment of initial temperature T_s

Only the initial temperature was changed, and other parameters were kept constant and assigned values. The number of iterations was set to 100, the cooling coefficient was 0.9, and the end temperature was 0.02. Twelve groups of temperatures were selected, and the simulated annealing program was run 10 times at each temperature to reduce the accidental error. Table 2 shows the data of the optimization process.

Table 2. Program optimization at 1000 degrees centigrade.

$T_s = 1000$	Traversal Distance Optimized (m)	Running Time (s)	Cooling Iteration Times
1	18.7174	4.1342	32.0000
2	20.6193	3.8558	21.0000
3	20.9443	3.4763	9.0000
4	19.1814	4.1061	29.0000
5	19.9797	3.9653	23.0000
6	20.3760	4.2626	32.0000
7	20.4530	4.3658	36.0000
8	21.6306	3.8912	21.0000
9	20.0824	3.9527	23.0000
10	19.6808	4.2147	31.0000
MIN	18.7174	3.4763	9.0000
AVG	20.1665	4.0225	25.7000
Standard deviation	0.8451	0.2546	7.9029

The average traversal distance and the minimum value at each temperature, as well as the standard deviation, were counted to generate the line chart (Figure 16). The blue line is the average traversal distance at each temperature, which was utilized to evaluate the overall optimization performance of the algorithm. The smaller the average traversal distance, the better the performance of the algorithm at that temperature. The yellow

line represents the optimized minimum value at each temperature, which represents the ability to find the optimal solution at each set of temperatures. The smaller the minimum value, the stronger mining ability of the algorithm at that temperature. The green line represents the standard deviation of each set of data, which was utilized to judge the degree of discreteness of a set of data. If the standard deviation was larger, it means that the set of data fluctuated greatly, and the performance of the algorithm was unstable. The abscissa represents the temperature of each group, and the ordinate represents the corresponding distance at a certain temperature. According to Figure 16, when the initial temperature was 100 °C, the average traversal distance and the minimum distance were the largest, indicating that the performance of the algorithm was poor. As the temperature increased, the average traversal distance and the minimum distance decreased. After the initial temperature reached 1500 °C, the decrease in amplitude of both slowed down as the initial temperature increased. After the initial temperature reached 3000 °C, the average traversal distance and minimum distance remained basically unchanged, and the algorithm performance tended to be stable. At the same time, it can be seen that the standard deviation fluctuated up and down at the beginning and then decreased to zero at the end, indicating that the algorithm optimization performance became more stable. In summary, 3000 °C was selected as the initial temperature of the algorithm to ensure the performance and reduce the waste of time.

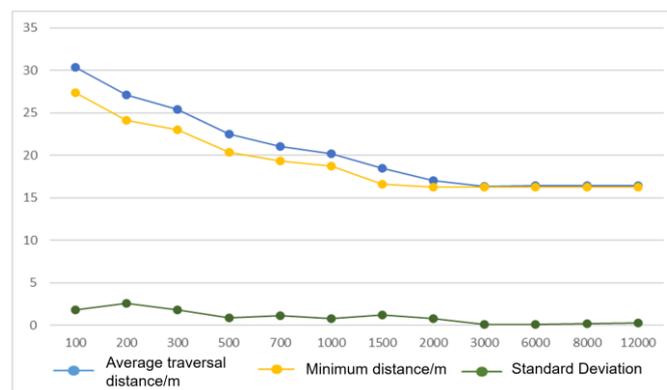


Figure 16. Optimization effect at different temperatures.

(2) The adjustment of the number of inner loop iteration M

Only the iteration number of inner loop was changed, and other parameters were kept constant and assigned. The initial temperature was set to 3000, the cooling coefficient was 0.9, and the end temperature was 0.02. Twelve groups of iteration numbers were selected, and the simulated annealing program was run 10 times at each iteration number to reduce the accidental error (Table 3).

Table 3. Program optimization under 100 loop times.

$M = 100$	Traversal Distance Optimized (m)	Running Time (s)	Cooling Iteration Times
1	16.3104	5.8098	83
2	17.5659	7.2418	119
3	16.2374	4.2802	32
4	16.3471	5.5327	60
5	16.8163	7.1329	122
6	16.5062	8.8083	171
7	16.334	6.6753	108
8	17.3419	7.0707	122
9	16.7095	8.8845	178
10	16.8789	8.5171	166
MIN	16.2374	4.2802	32
AVG	16.70476	6.99533	116.1
Standard deviation	0.4557	1.5009	48.0681

The average traversal distance, minimum value, and standard deviation under each internal iteration number were counted, and the trend line chart was drawn. When the number of internal iterations was too small, the optimization ability of the algorithm was limited, and the optimization effect of the average traversal distance was not obvious (Figure 17). With the increase in the number of internal iterations, the average traversal distance and the optimized minimum distance both decreased, indicating that the optimization ability of the algorithm was enhanced with the increase in the number of internal iterations. As the number of internal iterations increased, both the average traversal distance and the optimal minimum distance decreased, indicating that the optimization ability of the algorithm increased as the number of internal iterations increased. When the number of iterations reached 150, the performance of the algorithm almost did not improve with the increase in the iteration numbers. Therefore, the algorithm was already at a relatively optimal level at this time, and further increasing the number of internal iterations would only increase the time cost. In summary, the number of inner iterations was chosen to be 150.

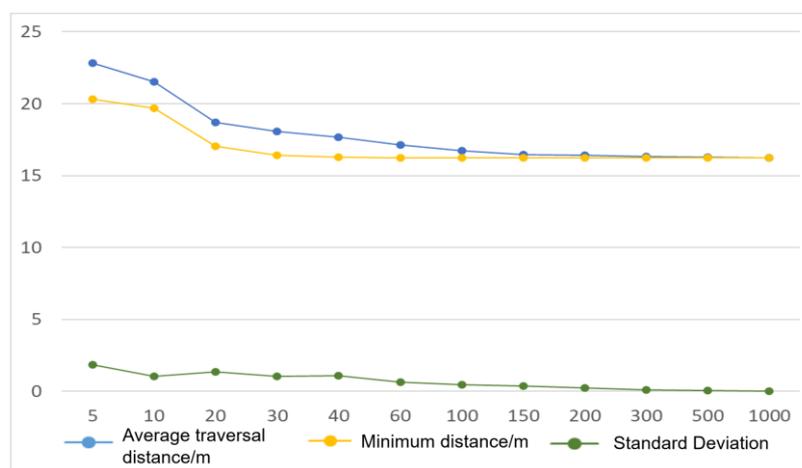


Figure 17. Optimization effect under different iteration numbers.

(3) The adjustment of cooling coefficient α

Only the cooling coefficient was changed, and other parameters were kept constant and assigned. The initial temperature was set to 3000, the iteration number of each temperature was 150, and the end temperature was 0.02. Twelve groups of cooling coefficient were selected, and the simulated annealing program was run 10 times at each cooling coefficient to reduce the accidental error (Table 4).

Table 4. Program optimization under 0.99 cooling coefficient.

$\alpha = 0.99$	Traversal Distance Optimized (m)	Running Time (s)	Cooling Iteration Times
1	16.2242	8.1109	107
2	16.2374	5.6527	55
3	16.2242	5.0363	43
4	16.3104	9.1028	133
5	16.2910	8.3292	113
6	16.2242	6.4719	71
7	16.2374	5.3622	51
8	16.2374	4.8088	37
9	16.2374	3.575	10
10	16.4018	7.1986	89
MIN	16.2242	3.575	10
AVG	16.26254	6.36484	70.9
Standard deviation	0.0571	1.7823	38.7941

The average traversal distance and the minimum value, as well as the standard deviation, at each iteration number were counted to generate the trend line chart (Figure 18). The blue line is the average traversal distance at each temperature. The yellow line represents the optimized minimum value at each temperature. The green line represents the standard deviation of each set of data. The abscissa represents the cooling coefficient of each group, and the ordinate represents the corresponding distance at a certain cooling coefficient. According to Figure 18, with the increase in the cooling coefficient, the optimization ability of the algorithm became better. When the cooling coefficient reached 0.99, with the increase in the cooling coefficient, the optimized effect of the simulated annealing algorithm was not obvious, and the optimized average traversal distance and the optimized minimum value of each group tended to be consistent. Therefore, 0.99 was an appropriate parameter value for this algorithm. In summary, the cooling coefficient of the algorithm was 0.99.

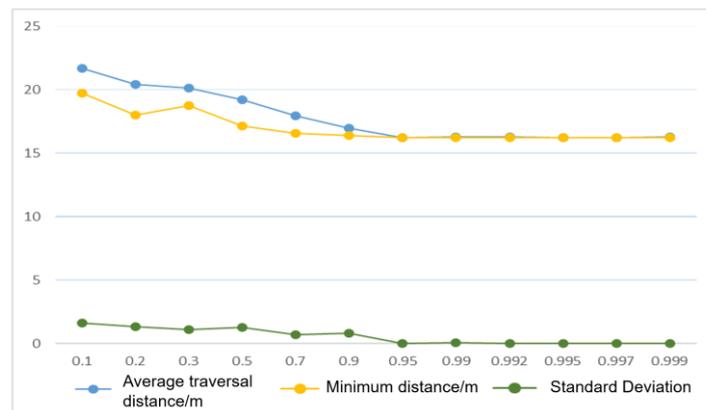


Figure 18. Optimization effect under different cooling coefficients.

Taking an area as an example, 20 points were randomly generated within the picking range of a robot arm to simulate the distribution of pears on a trellis. Since the impact in the vertical direction is small, only the coordinate information in X and Y directions was considered. Figure 19 separately shows that the randomly traversed path before optimization was disorderly and the picking path after optimization formed a neat loop, for which the optimization effect was obvious (Figure 19).

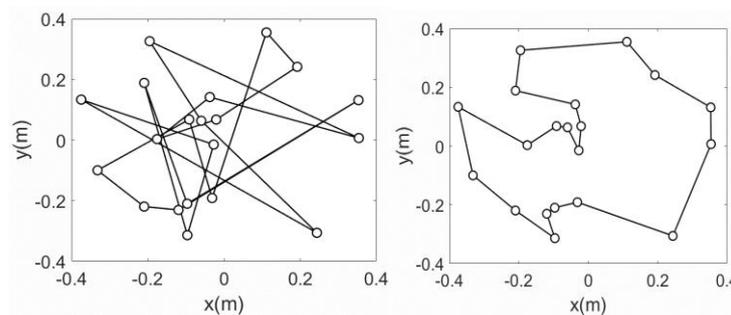


Figure 19. Comparison before and after the optimization of the simulated annealing algorithm.

The total number of iterations was 300. As the number of iterations increased, the total path length gradually decreased. In the range of 0–50 times, the curve was steeper, and the convergence speed was faster. At this time, the optimization effect was better. In the range of 50–120 times, the curve was a straight line, and the algorithm fell into a local optimum. As the number of iterations increased, the perturbation caused the solution to jump out of the current local optimum. After the number of iterations reached 240, a better result was achieved (Figure 20).

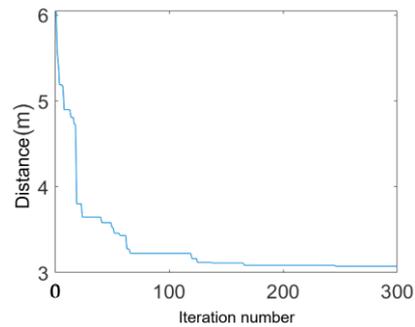


Figure 20. Iteration optimization of a simulated annealing algorithm.

5. Picking Experiment

5.1. Picking Experiment with Different End-Pickers

In this experiment, different numbers of fruits were set to test the picking effect of different end-pickers. Each set of fruits had the same arrangement position, the same picking sequence, and the same number. The picking experiment was carried out using the gripper and the pick-place integrated end-picker. In addition, all the fruits in this experiment were models, so all the cutting action of fruit stems was a simulation. The picking pose of the end-picker was vertically up (Figure 21b).



Figure 21. Picking diagram. (a) Initial pose; (b) target pose.

The binocular camera at the bottom of the robot arm obtained the three-dimensional spatial coordinates of the fruit. After the coordinate transformation, the recognized fruit coordinates were transmitted to the robot arm. The flexible gripper picking process is shown in Figure 22. The pick-place integrated end-picker moved from the initial pose (Figure 23a) to the lower part of target position (Figure 23b), and it continued to move to the target position (Figure 23c). The mechanical arm carried the end-picker close to the fruit, setting the fruit into the cavity. At this time, the steering engine drove the cutting mechanism to close through the reduction gear and clamp the pear stem (Figure 23d). Then, the robot arm drove the cutting mechanism to move down and cut off the pear stem, and the fruit from the cavity fell into the collection tube in the direction of the arrow, and through the collection tube entered the storage basket, which completed the picking and collection (Figure 23e). Finally, after completing a picking task, the cutting mechanism was driven to open by the steering engine (Figure 23f).

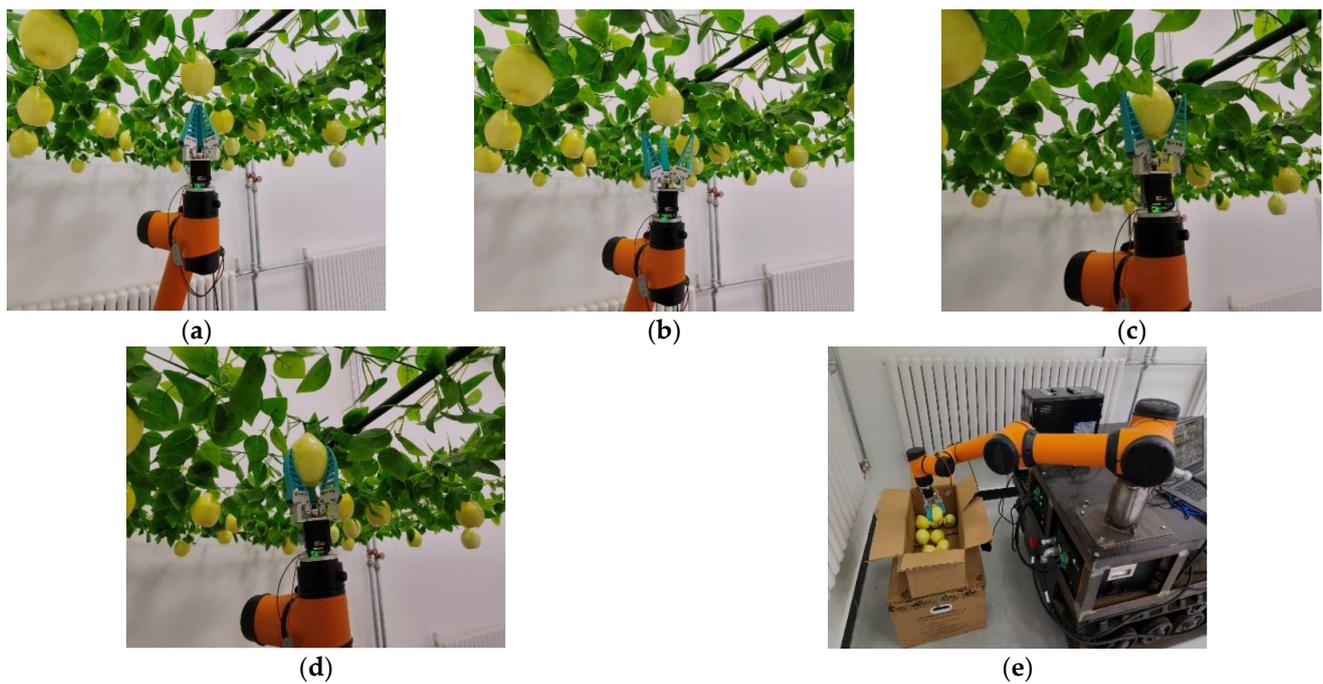


Figure 22. The process of flexible grippers from picking fruits to placing them in the basket. (a) Reaching below the target position; (b) gripper open; (c) reaching the target location; (d) gripper closed; (e) putting the pear in the basket.

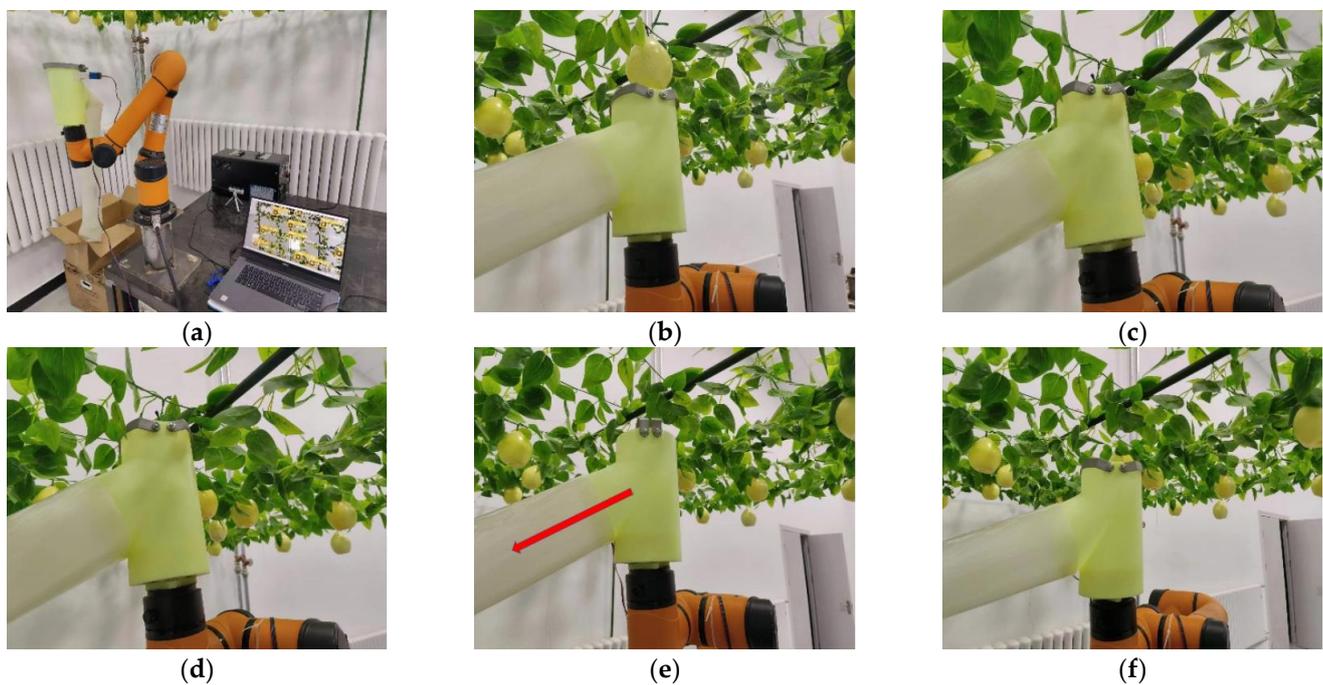


Figure 23. The integration process of the pick-place integrated end-picker from the initial pose to the completion of picking. (a) Initial pose; (b) reaching below the target position; (c) reaching the target location; (d) cutting mechanism closed; (e) cutting completed; (f) cutting mechanism open.

Different numbers of pears were randomly placed within the picking range of the robot arm. The picking experiments of two different end-pickers are shown in Table 5 below:

Table 5. Picking experimental data statistics.

Fruit Number	Picking Time (s)		Percent (%)	Picking Failure Number	
	Grippers	Pick-Place Integrated		Grippers	Pick-Place Integrated
2	35.32	21.96	37.83	0	0
4	69.65	45.59	34.54	1	1
6	97.65	61.28	37.24	2	1
8	138.35	81.09	41.39	2	0
10	165.94	111.63	32.73	3	2

It can be seen from the table that when picking with flexible claws, the picking time of each fruit was about 16 s, while the pick-place integrated end-picker took about 10 s to pick one fruit. The efficiency of the pick-place integrated end-picker was increased by approximately 37%. At the same time, the probability of picking failure using a flexible gripper was higher than that of the pick-place integrated end-picker. When the number of fruits was ten, the number of failed gripper pickings was three, for which the failure probability was 30%. Conversely, the failure probability of the integrated end-picker was only 20%. In several sets of experiments, the overall success probability of the integrated end-picker was about 86.67%, while that of the gripper was only 73.33%.

5.2. Comparison before and after Picking Sequence Optimization

In this experiment, different numbers of fruits were set to test the picking effect of the single robot arm system, and the picking sequence was optimized by the simulated annealing algorithm. The experiment used the integrated end-picker, for which each set of fruits was randomly distributed within the picking range of the robot arm, and the fruit distribution position was consistent before and after optimization. The picking experiment result was as shown in Table 6 below:

Table 6. Experiment result before and after optimization.

Fruit Number	Picking Time (s)		Percent (%)	Picking Failure Number	
	Random Traversal	Simulated Annealing		Random Traversal	Simulated Annealing
2	11.94	12.14	-1.68	0	0
4	28.65	23.51	17.94	0	1
6	41.59	31.24	24.89	1	0
8	50.98	39.61	22.30	2	2
10	63.43	46.58	26.56	2	1

It can be seen from the table that picking efficiency was improved by about 20% after the optimization of the simulated annealing algorithm, for which the improvement efficiency also increased with the number of fruits. The picking route was messy and lengthy before optimization, and the traversal distance was greatly reduced after optimization by the simulated annealing algorithm (Figure 24).

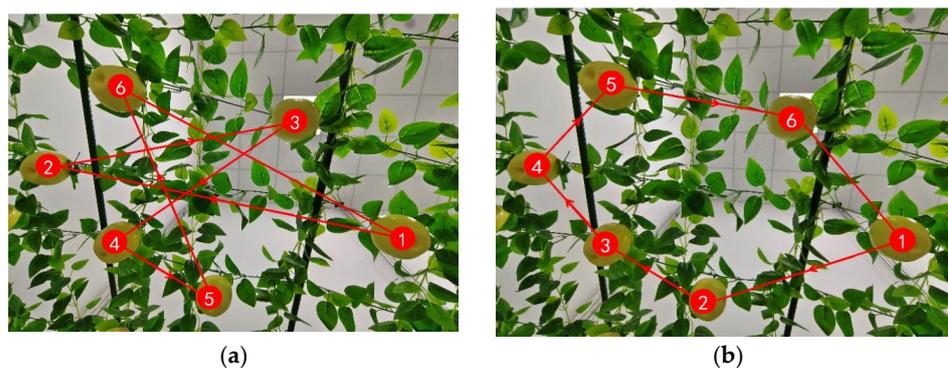


Figure 24. Comparison before and after optimization. (a) Before optimization; (b) after optimization.

5.3. Simulation of Task Allocation for Multiple Robotic Arms

We randomly generated 5, 10, 20, 30, 40, and 50 points in the picking plane area of the two robot arms to simulate the growth distribution position of pears, and we counted the number of fruits falling into the three areas and then followed the proposed task allocation method of multi robotic arms allocating picking tasks to corresponding robot arms. Finally, we counted the task volume of each robot arm. From Table 7, it can be seen from the six sets of data that the task allocation method proposed in this paper was able to effectively improve the utilization rate of a robot arm and make the task volume of the multi-machine system tend to be evenly distributed.

Table 7. Task allocation for two robot arms.

Group	Fruits Number in Region			Task Volume of the Robotic Arm	
	P_1	P_2	P_3	R_1	R_2
5	2	0	3	2	3
10	5	3	2	5	5
20	5	12	3	8	12
30	17	9	4	17	13
40	21	14	5	21	19
50	26	16	8	26	24

We randomly generated 10, 20, 30, 40, 50, and 60 points in the working area of three robot arms to simulate the growth distribution position of pears, and then we counted the number of fruits falling into the seven areas, and finally, we allocated tasks. It can be seen from Table 8 that for the three-robot-arm system, taking the number of fruits as 40 as an example, after using the task allocation method proposed in this article, the task volume of the three robot arms were 12, 14, and 14 each. The minimum and maximum of robot arm task volume differed by two fruits.

Table 8. Task allocation for three robot arms.

Group	Fruit Number in the Region							Task Volume of the Robotic Arm		
	P_1	P_2	P_3	P_4	P_5	P_6	P_7	R_1	R_2	R_3
10	3	1	0	1	2	1	2	3	4	3
20	4	3	5	1	2	2	3	6	6	8
30	4	6	7	4	3	2	4	12	9	9
40	8	9	7	4	3	4	5	12	14	14
50	9	7	11	8	5	7	3	16	18	16
60	11	13	10	6	6	7	7	17	20	23

In summary, the task allocation method for the multiple robotic arm system proposed in this paper is effective and avoids the unreasonable task allocation that causes some of the robot arms to have a particularly large task volume and work all the time, while other robot arms have a particularly small task volume and a short working time, improving the utilization rate of each robot arm and the overall picking efficiency.

5.4. Simulation of Two Robot Arms

We exported the URDF file of the robot arm and the STL model in SOLIDWORKS, and we imported the file into Simulink. The two robot arms were 100 cm apart, and the visualization is shown in Figure 25a.

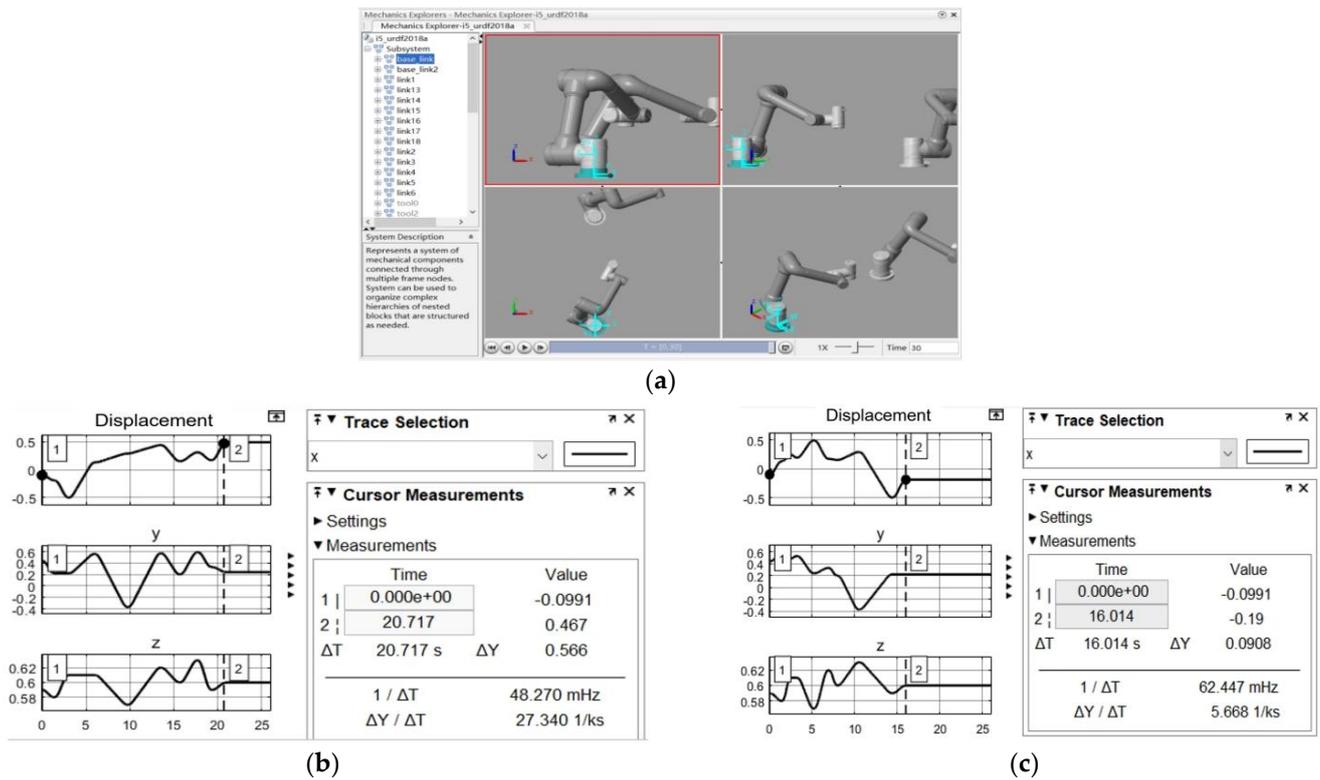


Figure 25. Time record with different algorithms. (a) Simulation visualization diagram; (b) random traversal; (c) simulated annealing.

A total of 4, 6, 8, and 10 points were randomly generated in the picking range of the two robot arms to simulate the position of the pear, and the proposed task allocation method was used to allocate the tasks of the two robot arms. The manipulator directly moved to the next target point after reaching a target point from the initial pose, so only the movement process time of the manipulator was recorded. The movement time was read from the oscilloscope as shown in Figure 25b,c, and the time data are shown in Table 9.

Table 9. Traverse time of two robot arms.

Group	Random Traversal (s)			Simulated Annealing (s)		
	R ₁	R ₂	Overall System	R ₁	R ₂	Overall System
4	13.854	11.651	13.854	10.533	9.032	10.533
6	15.217	18.965	18.965	10.657	14.367	14.367
8	21.717	25.361	25.361	16.014	18.549	18.549
10	30.941	25.758	30.941	23.187	19.552	23.187

As can be seen from Table 9, there was little difference in picking time between the two robot arms in each group after task assignment, and the picking time of the overall system was determined by the robot arm that spent the longest time. The time of traversing all points was reduced by about 20% by simulated annealing.

6. Conclusions

In view of the current inefficient work of picking robots, this paper studied the collaborative picking of multiple robotic arms in a trellised pear orchard environment to improve the picking efficiency of the system. The physical characteristics of pears were studied through experiments, and the structural design of the end-picker was carried out based on this. We constructed the private dataset of pears, and finally, we output the three-dimensional coordinates of the target fruit through the Yolo-v5s detection model combined with the depth camera and carried out the object detection through the trained

weight, and following this, we completed the object detection task accurately. The camera calibration and hand–eye calibration were completed, and the images of the calibration plate at different angles and distances were collected with the camera. After inputting into the Matlab calibration toolbox, the camera’s distortion coefficient and internal parameter matrix were obtained, for which the camera calibration was completed. Through the relationship between two fixed coordinate systems, twenty points were selected to form a hyperparameter equation, and finally, the hand–eye calibration matrix was obtained by fitting the results using the least squares method. The conversion from the pixel coordinate system to the robot arm base coordinate system was completed with good accuracy.

A task allocation method for a multiple robotic arm system was proposed, and the picking sequence was optimized through the simulated annealing algorithm. The several key parameters of the simulated annealing algorithm on the algorithm performance were studied, and the optimal parameter values were selected. After optimization, the picking efficiency was clearly improved. The final result was as follows: initial temperature $T_s = 3000$ °C, internal loop iteration number $M = 150$, and cooling coefficient $\alpha = 0.99$.

The experimental results showed that the picking efficiency of the designed pick-place integrated end-picker is higher than that of the traditional gripper. The success rate of the picking mechanism designed in this paper was 86.67%, which is about 30% higher than that of the claw-gripper. The task allocation method proposed in this paper can make the task volume of a multiple-robotic-arm system tend to be evenly divided, and it obviously improved the utilization rate of each robot arm. Through the simulated annealing algorithm, compared with random traversal, the efficiency of the optimized picking path was increased by about 20%. Moreover, with the increase in fruit number, the efficiency showed an increasing trend.

However, this paper only focused on the structural design of the end-picker, and it did not conduct real experiments under different lighting conditions. In the next step, we will further develop the vision system suitable for a wider range of application scenarios and develop supporting devices that can reduce recognition interference such as light source occlusion, so as to further improve the accuracy rate of the target detection system and the success rate of the picking system. In addition, on the basis of studying the agronomic characteristics and the damage mechanism of fruits, we will further improve the mechanical structure and matching algorithm of the picker to achieve a better picking effect, which is also the next goal of our research.

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

Funding: This work was funded by the National Natural Science Foundation of China under grant 32372592; CCF- Baidu Apollo Joint Development Project Fund.

Data Availability Statement: Data are contained within the article.

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

References

1. Mu, L.; Cui, G.; Liu, Y.; Cui, Y.; Fu, L.; Gejima, Y. Design and simulation of an integrated end-effector for picking kiwifruit by robot. *Inf. Process. Agric.* **2020**, *7*, 58–71. [[CrossRef](#)]
2. Yang, C.H.; Xiong, L.Y.; Wang, Z.; Wang, Y.; Shi, G.; Kuremot, T.; Zhao, W.H.; Yang, Y. Integrated detection of citrus fruits and branches using a convolutional neural network. *Comput. Electron. Agric.* **2020**, *174*, 105469. [[CrossRef](#)]
3. Yang, Q.; Rong, J.; Wang, P.; Yang, Z.; Geng, C. Real-time detection and localization using SSD method for oyster mushroom picking robot. In Proceedings of the 2020 IEEE International Conference on Real-Time Computing and Robotics (RCAR), Asahikawa, Japan, 28–29 September 2020; pp. 158–163.

4. Kang, H.; Zhou, H.; Chen, C. Visual perception and modeling for autonomous apple harvesting. *IEEE Access* **2020**, *8*, 62151–62163. [[CrossRef](#)]
5. Peng, H.; Xue, C.; Shao, Y.; Chen, K.; Xiong, J.; Xie, Z.; Zhang, L. Semantic segmentation of litchi branches using DeepLabV3+ model. *IEEE Access* **2020**, *8*, 164546–164555. [[CrossRef](#)]
6. Li, J.; Tang, Y.; Zou, X.; Lin, G.; Wang, H. Detection of fruit-bearing branches and localization of litchi clusters for vision-based harvesting robots. *IEEE Access* **2020**, *8*, 117746–117758. [[CrossRef](#)]
7. Liang, C.; Xiong, J.; Zheng, Z.; Zhong, Z.; Li, Z.; Chen, S.; Yang, Z. A visual detection method for nighttime litchi fruits and fruiting stems. *Comput. Electron. Agric.* **2020**, *169*, 105192. [[CrossRef](#)]
8. Suo, R.; Gao, F.; Zhou, Z.; Fu, L.; Song, Z.; Dhupia, J.; Li, R.; Cui, Y. Improved multi-classes kiwifruit detection in orchard to avoid collisions during robotic picking. *Comput. Electron. Agric.* **2021**, *182*, 106052. [[CrossRef](#)]
9. Xiong, Y.; Ge, Y.; From, P.J. An improved obstacle separation method using deep learning for object detection and tracking in a hybrid visual control loop for fruit picking in clusters. *Comput. Electron. Agric.* **2021**, *191*, 106508. [[CrossRef](#)]
10. Li, X.; Pan, J.; Xie, F.; Zeng, J.; Li, Q.; Huang, X.; Liu, D.; Wang, X. Fast and accurate green pepper detection in complex backgrounds via an improved Yolov4-tiny model. *Comput. Electron. Agric.* **2021**, *191*, 106503. [[CrossRef](#)]
11. Wang, L.; Zhao, Y.; Liu, S.; Li, Y.; Chen, S.; Lan, Y. Precision detection of dense plums in orchards using the improved YOLOv4 model. *Front. Plant Sci.* **2022**, *13*, 839269. [[CrossRef](#)]
12. Yan, B.; Fan, P.; Lei, X.; Liu, Z.; Yang, F. A real-time apple targets detection method for picking robot based on improved YOLOv5. *Remote Sens.* **2021**, *13*, 1619. [[CrossRef](#)]
13. Xiong, Y.; Ge, Y.; From, P.J. An obstacle separation method for robotic picking of fruits in clusters. *Comput. Electron. Agric.* **2020**, *175*, 105397. [[CrossRef](#)]
14. Xiong, Y.; Peng, C.; Grimstad, L.; From, P.J.; Isler, V. Development and field evaluation of a strawberry harvesting robot with a cable-driven gripper. *Comput. Electron. Agric.* **2019**, *157*, 392–402. [[CrossRef](#)]
15. Ye, L.; Duan, J.; Yang, Z.; Zou, X.; Chen, M.; Zhang, S. Collision-free motion planning for the litchi-picking robot. *Comput. Electron. Agric.* **2021**, *185*, 106151. [[CrossRef](#)]
16. Feng, Q.; Zou, W.; Fan, P.; Zhang, C.; Wang, X. Design and test of robotic harvesting system for cherry tomato. *Int. J. Agric. Biol. Eng.* **2018**, *11*, 96–100. [[CrossRef](#)]
17. Oktarina, Y.; Dewi, T.; Risma, P.; Nawawi, M. Tomato harvesting arm robot manipulator; a pilot project. *J. Phys. Conf. Series.* **2020**, *1500*, 012003. [[CrossRef](#)]
18. Wei, J.; Yi, D.; Bo, X.; Guangyu, C.; Dean, Z. Adaptive variable parameter impedance control for apple harvesting robot compliant picking. *Complexity* **2020**, *2020*, 1–15. [[CrossRef](#)]
19. Liu, C.-H.; Chiu, C.-H.; Chen, T.-L.; Pai, T.-Y.; Chen, Y.; Hsu, M.-C. A soft robotic gripper module with 3D printed compliant fingers for grasping fruits. In Proceedings of the 2018 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Auckland, New Zealand, 9–12 July 2018; pp. 736–741.
20. Pi, J.; Liu, J.; Zhou, K.; Qian, M. An octopus-inspired bionic flexible gripper for apple grasping. *Agriculture* **2021**, *11*, 1014. [[CrossRef](#)]
21. Hohimer, C.J.; Wang, H.; Bhusal, S.; Miller, J.; Mo, C.; Karkee, M. Design and field evaluation of a robotic apple harvesting system with a 3D-printed soft-robotic end-effector. *Trans. ASABE* **2019**, *62*, 405–414. [[CrossRef](#)]
22. Yu, X.; Fan, Z.; Wang, X.; Wan, H.; Wang, P.; Zeng, X.; Jia, F. A lab-customized autonomous humanoid apple harvesting robot. *Comput. Electr. Eng.* **2021**, *96*, 107459. [[CrossRef](#)]
23. Arima, S.; Kondo, N.; Yagi, Y.; Monta, M.; Yoshida, Y. Harvesting robot for strawberry grown on table top culture, 1: Harvesting robot using 5 DOF manipulator. *J. Soc. High Technol. Agric.* **2001**, *13*, 159–166. [[CrossRef](#)]
24. Arima, S.; Monta, M.; Namba, K.; Yoshida, Y.; Kondo, N. Harvesting robot for strawberry grown on table top culture (part 2) harvesting robot with a suspended manipulator under cultivation bed. *Shokubutsu Kojo Gakkaishi* **2003**, *15*, 162–168. [[CrossRef](#)]
25. Van Henten, E.J.; Schenk, E.J.; Van Willigenburg, L.G.; Meuleman, J.; Barreiro, P. Collision-free inverse kinematics of the redundant seven-link manipulator used in a cucumber picking robot. *Biosyst. Eng.* **2010**, *106*, 112–124. [[CrossRef](#)]
26. Bontsema, J. Picking robot for peppers (interview met Jan Bontsema). *WageningenWorld* **2011**, *2011*, 6.
27. Kahani, A. Multi-Robot Crop Harvesting Machine. U.S. Patent 9,475,189, 25 October 2016.
28. Williams, H.A.; Jones, M.H.; Nejati, M.; Seabright, M.J.; Bell, J.; Penhall, N.D.; Barnett, J.J.; Duke, M.D.; Scarfe, A.J.; Ahn, H.S.; et al. Robotic kiwifruit harvesting using machine vision, convolutional neural networks, and robotic arms. *Biosyst. Eng.* **2019**, *181*, 140–156. [[CrossRef](#)]
29. Fu, L.; Tola, E.; Al-Mallahi, A.; Li, R.; Cui, Y. A novel image processing algorithm to separate linearly clustered kiwifruits. *Biosyst. Eng.* **2019**, *183*, 184–195. [[CrossRef](#)]
30. Trinidad, J.B. Machine for Automatically Harvesting Fruit Cultivated in Rows. U.S. Patent 13/124,577, 20 October 2011.

31. Pitzer, R. Automated Selective Harvesting of Crops with CONTINUOUS offload. U.S. Patent 9,554,513, 31 January 2017.
32. Guillén-Navarro, M.A.; Martínez-España, R.; Bueno-Crespo, A.; Morales-García, J.; Ayuso, B.; Cecilia, J.M. A decision support system for water optimization in anti-frost techniques by sprinklers. *Sensors* **2020**, *20*, 7129. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.