

## Target Positioning and Sorting Strategy of Fruit Sorting Robot Based on Image Processing

Jianhong Yu, Weijie Miao, Guangben Zhang, Kai Li, Yinggang Shi\*, Li Liu

College of Mechanical and Electronic Engineering, Northwest A&F University, Yangling 712100, China

Corresponding Author Email: [syg9696@nwafu.edu.cn](mailto:syg9696@nwafu.edu.cn)



<https://doi.org/10.18280/ts.380326>

### ABSTRACT

**Received:** 15 February 2021

**Accepted:** 6 May 2021

#### Keywords:

*three-dimensional (3D) scene object recognition, fruit sorting, industrial robot, recognition of fruit maturity*

To a certain extent, automated fruit sorting systems reflect the degree of automated production in modern food industry, and boast a certain theoretical and application value. The previous studies mostly concentrate on the design of robot structure, and the control of robot motions. There is little report on the feature extraction of fruits in specific applications of fruit sorting. For this reason, this paper explores the target positioning and sorting strategy of fruit sorting robot based on image processing. Firstly, the authors constructed a visual sorting system for fruit sorting robot, and explained the way to recognize objects in three-dimensional (3D) scene and to reconstruct the spatial model based on sorting robot. Next, the maturity of the identified fruits was considered the prerequisite of dynamic sorting of fruit sorting robot. Finally, the program flow of the fruit sorting robot was given. The effectiveness of our strategy was verified through experiments.

## 1. INTRODUCTION

With the advancement of intelligent manufacturing and industrial robot technology, the application of industrial robots has gradually alleviated the hard problem of sorting a huge number of various types of fruits and vegetables [1-5]. Based on machine vision, the automated fruit sorting system reflect the degree of automated production in modern food industry, and provide reasonable application scenarios for the development of advanced science and technology, such as visual recognition and image processing. Therefore, the automated fruit sorting system boasts certain theoretical and application value.

The position and angle sensors of robots have errors in information acquisition. Sidehabi et al. [6] constructed a relative motion model through distance and angle-based particle filtering, which enhances the positioning accuracy of robot sorting robot for moving targets. Based on power transform method, Khoje [7] improved the working environment image and spatial contrast of fruit sorting robot, selected the optimal threshold for working environment image segmentation, using between-class variance, and established a robot target positioning model based on deep learning network. Wahyuni and Affan [8] sets up rules for target recognition and positioning of fruit sorting robot, proposed a target recognition algorithm that conforms to the color and shape features of the fruits, and applied Kinect and laser sensors to acquire data over a long distance.

To recognize and capture objects, the robot needs to adapt to the complex and changeable unstructured environment [9-12]. Apte and Patavardhan [13] developed a target recognition and control method for industrial robots based on binocular vision and color segmentation, proposed a Denavit-Hartenberg (D-H) model to compute the control signals of the three-dimensional (3D) coordinates for the robot, and eventually completed the recognition and grabbing task.

During high-speed point-to-point motions, traditional

sorting robots face limitations like vibration, discontinuity, and low fault tolerance [14-19]. To improve the speed and stability of robot operations, Raka et al. [20] established the fitness function through particle swarm optimization (PSO), and formulated a visual servo control system for fruit sorting robot, which does not include the Jacobian matrix of target image and the calibration process.

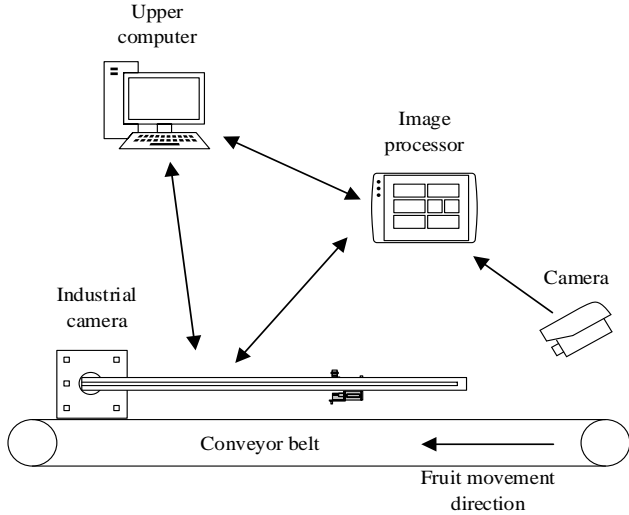
Traditional visual algorithms could hardly classify and position various types of fruits with complex shapes [21-26]. The previous studies mostly concentrate on the design of robot structure, and the control of robot motions. There is little report on the feature extraction of fruits in specific applications of fruit sorting.

For this reason, this paper explores the target positioning and sorting strategy of fruit sorting robot based on image processing. Section 2 constructs a visual sorting system for fruit sorting robot, and explained the way to recognize objects in 3D scene and to reconstruct the spatial model based on sorting robot. Section 3 gives the program flow of the fruit sorting robot, and highlights the maturity of the identified fruits as the prerequisite of dynamic sorting of fruit sorting robot. Experimental results confirm the effectiveness of our strategy.

## 2. POINT CLOUD DATA PROCESSING AND FRUIT RECOGNITION

Target positioning, as the most critical technology in the working process of fruit sorting robot, involves the identification of target fruits, and the solving of grabbing poses. 3D data processing is increasingly mature, as 3D cameras are being applied to wider fields. Relevant scholars at home and abroad have shifted their focus to the object recognition and spatial model reconstruction for the sorting robot in the 3D scene. This paper mainly aims to recognize and sort fruits that are randomly distributed. Figure 1 is a sketch map of the visual

sorting system of fruit sorting robot. In the system, the fruit surface is described by the 3D point cloud data captured by a Kinect camera, and processed with a series of 3D data processing algorithms. In this way, the maturity, size, and other attributes of fruits become more authentic. Eventually, the relevant information is converted into the pose information that can be recognized by the robot, which enables the robot to complete fruit sorting.



**Figure 1.** Visual sorting system of fruit sorting robot

Surface normal and curvature are the core features of 3D point cloud data. Surface normal facilitates the calculation of the feature descriptors about the geometric properties of the target. Hence, solving the surface normal of the target is equivalent to obtaining the eigenvectors and eigenvalues from the covariance matrix, which is created from the neighborhood elements of the query point in the paired least squares fitting plane. Let  $L$  be the number of neighboring points of point  $O_i$ ;  $O^*$  be the 3D centroid of the nearest neighbors. Then, the covariance matrix  $D$  corresponding to each point  $O_i$  can be described by:

$$D = \frac{1}{l} \sum_{i=1}^l (O_i - O^*) \cdot (O_i - O^*)^{\phi} \quad (1)$$

Let  $\mu_i$  and  $\hat{u}_i$  be the  $i$ -th eigenvalue and eigenvector of the covariance matrix, respectively. Then,  $D$  needs to satisfy:

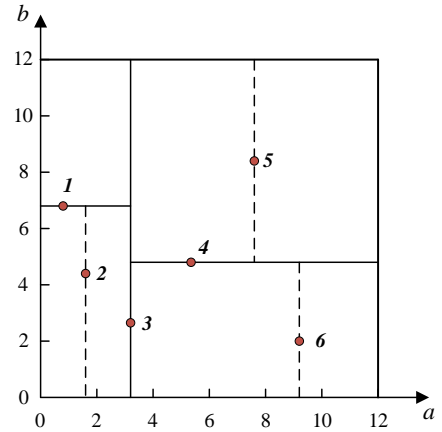
$$D \cdot \hat{u}_j = \mu_j \cdot \hat{u}_j \quad (2)$$

In our application scenario, the surface information of fruits is demonstrated by the 3D point cloud image data collected by the Kinect camera, while the surface bending of fruits is characterized by curvature. In formula (2), parameter  $j$  could take the value of 1, 2, or 3. Thus, the eigenvalue  $\mu_i$  of covariance matrix  $D$  can be described as  $\{\mu_1, \mu_2, \mu_3\}$ , where  $\mu_1 > \mu_2 > \mu_3$ . The curvature can be calculated by:

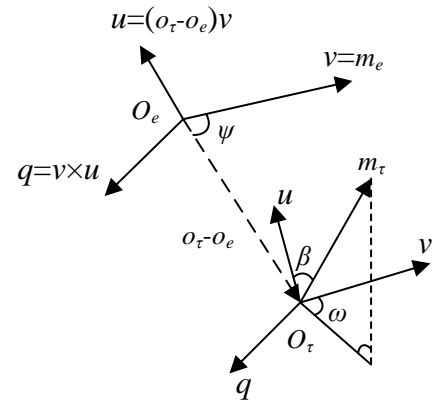
$$\xi = \frac{\mu_1}{\mu_1 + \mu_2 + \mu_3} \quad (3)$$

Before matching the 3D point cloud data and reconstructing the images of each fruit, it is necessary to extract the features from the point cloud data. In this paper, the maturity and size

of the fruit are recognized based on the features of point cloud data. Point feature histogram is a common local feature descriptor of point cloud data. The point feature histogram descriptor represents the geometric features of the fruit according to the relationship between the normal of a point in the point cloud data and that of the points in its neighborhood. Here, the neighboring points are searched for by the  $k$ -dimensional (K-D) tree. The principle of the K-D tree is explained in Figure 2. This flexible tool is suitable for processing the small 3D point cloud data of a single fruit. Let  $O_w$  be the center point of geometric features. Then, the neighborhood calculated by the point feature histogram descriptor is a circular area with the radius of  $\rho$ . The histogram can be derived from the relationship between all neighboring points.



**Figure 2.** Index structure of K-D tree



**Figure 3.** A fixed local coordinate system

Figure 3 shows the relative location between any two points  $o_\tau$  and  $o_e$  in the point cloud. The relative deviations between the two points can be represented by the normal  $m_\tau$  and  $m_e$  of the two points. Next, a local coordinate system  $v-u-q$  was defined on one of the two points. Then, the three axes can be respectively expressed as:

$$v = m_e \quad (4)$$

$$u = v \times \frac{(o_\tau - o_e)}{\|o_\tau - o_e\|_2} \quad (5)$$

$$q = v \times u \quad (6)$$

The deviations between the two normal in coordinate system  $v$ - $u$ - $q$  can be described by:

$$\beta = u \cdot m_\tau \quad (7)$$

$$\psi = v \cdot \frac{o_\tau - o_e}{\|o_\tau - o_e\|_2} \quad (8)$$

$$\omega = \arctan(q \cdot m_\tau, v \cdot m_\tau) \quad (9)$$

The above analysis shows that, to obtain the point feature histogram descriptor, it is only necessary to compute the 4-tuple  $(\beta, \psi, \omega, r)$  of each point pair. The other parameters related to point cloud normal can be ignored. The descriptor adapts well to the sampling density or noise of the point cloud, and remains invariant to the rotation and translation of the point cloud surface.

The fast point feature histogram descriptor is even faster than point feature histogram descriptor. This descriptor has a simple calculation process: Firstly, the position relationship between every query point and its neighboring points is computed to obtain the corresponding point feature histogram. Next, the neighborhood is determined for each neighboring point. Let  $\omega_l$  be the distance between query point  $o_w$  and its neighboring point  $o_l$ . Then, the final histogram of  $O_w$  can be described by:

$$RO_F(O_w) = FO_S(O_w) + \frac{1}{l} \sum_{i=1}^l \frac{1}{\omega_i} \cdot FO_S(O_i) \quad (10)$$

The features extracted from fruit point cloud need to be matched with different fruit templates to complete fruit recognition. Most fruits are spherical or ellipsoidal. There are very minor differences if the fruits are observed from different angles. Therefore, it is easy for errors to occur in template matching. This calls for segmentation and classification of the features of fruit point cloud. This paper classifies fruit point cloud features with the radius-based surface descriptor, which is a global feature descriptor.

The radius-based surface descriptor relies on the semi-globally unique reference frame calculated for each region, and requires a histogram of cluster viewpoint features as the basis of analysis. Let  $P_i$  be the point cloud set of the continuous smooth area on fruit surface;  $N$  be the eigenvector of the weighted scatter matrix of the points in the set;  $p_i$  be the centroid of the smooth area;  $T$  be the maximum Euclidean distance between  $p_i$  and any point in  $P_i$ . Then, we have:

$$N = \frac{1}{\sum_{l \in P_i} (T - r_l)} \sum_{l \in P_i} (T - \|o_l - d_i\|_2) \quad (11)$$

$$(o_l - d_i)(o_l - d_i)^\phi$$

In the fruit sorting scene, if each descriptor is matched with each model descriptor in the fruit template point cloud library, then the coarse registration of the fruit point cloud features has been completed. Next, fine registration should be performed on fruit point cloud by the iterative closest point method, that is, a cloud point set should be matched with another cloud point set. Here, feature point matching can be viewed as a similarity retrieval problem between high-dimensional vectors by a distance function. Octree provides a nearest neighbor

search strategy for fine registration. The index structure of octree is illustrated in Figure 4. After the nearest neighbor search, the optimal coordinate transform between two-dimensional (2D) fruit image and 3D point cloud should be iteratively computed by the least squares method. The computation must ensure that the error function is smaller than the preset error. In this way, it is possible to capture the pose changes of the fruit relative to templates in the fruit sorting scene.

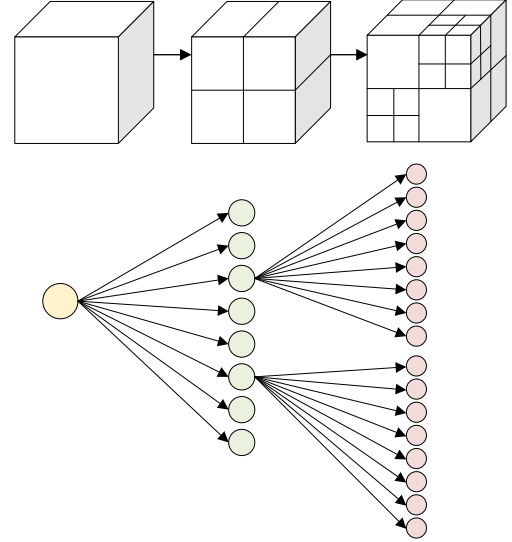


Figure 4. Index structure of octree

After coarse registration, the point cloud data  $O$  and  $O'$  can be expressed as:

$$O = \{o_1, \dots, o_m\}, O' = \{o'_1, \dots, o'_m\} \quad (12)$$

Let  $SP$  and  $TV$  be the rotation matrix and the translation vector. Then, the translation and radiation relationship can be established as:

$$o_i = SP \cdot o'_i + TV \quad (13)$$

The error term of the  $i$ -th point pair can be calculated by:

$$s_i = o_i - (SP \cdot o'_i + TV) \quad (14)$$

The least squares problem of point set to point set matching can be described by:

$$\min_{SP, TV} QG = \frac{1}{2} \sum_{i=1}^m \|(o_i - (SP \cdot o'_i + TV))\|_2^2 \quad (15)$$

The centroids of the two sets of point cloud data can be calculated by:

$$o = \frac{1}{m} \sum_{i=1}^m (o_i), o' = \frac{1}{m} \sum_{i=1}^m (o'_i) \quad (16)$$

After removing the centroid coordinates, the coordinates of each point in a point cloud can be described by:

$$w_i = o_i - o, w'_i = o'_i - o \quad (17)$$

In the optimal coordinate transform based on least squares method, the rotation matrix can be described by:

$$TV^* = o - SP \cdot o' \quad (19)$$

$$SP^* = \arg \min_{SP} \frac{1}{2} \sum_{i=1}^m \|w_i - SP \cdot w'_i\|^2 \quad (18)$$

### 3. SORTING STRATEGY BASED ON FRUIT MATURITY

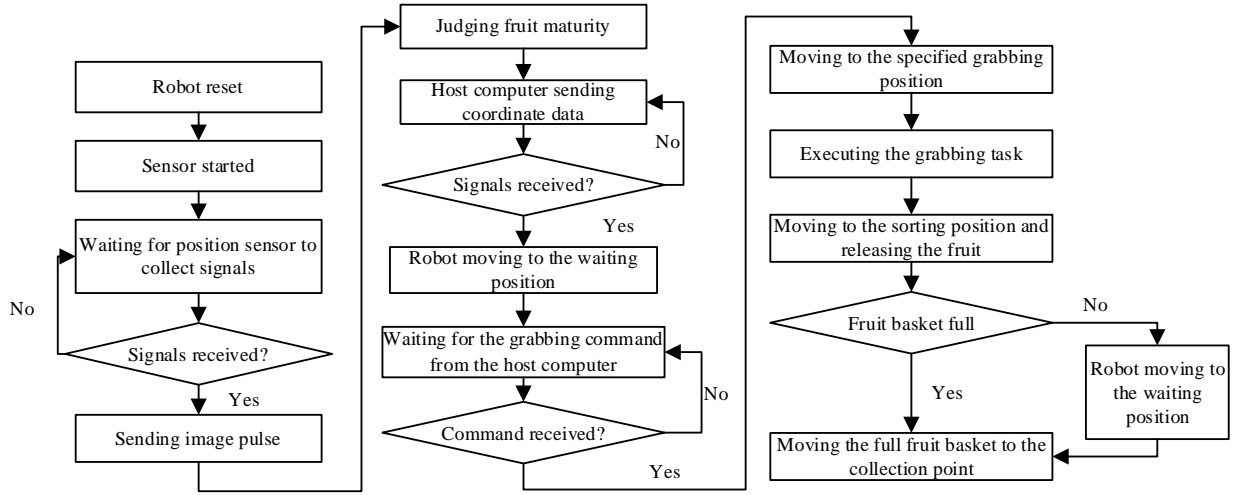


Figure 5. Program flow of fruit sorting robot

Figure 5 shows the program flow of the fruit sorting robot. To realize the dynamic fruit sorting, the robot must recognize the target fruit from the point cloud data, and evaluate the maturity of the recognized fruit before the host computer sends the coordinates of the fruit. The maturity evaluation is to clarify the correspondence between surface features (e.g., size, shape, and color) and fruit maturity. Based on machine vision, image processing technology can be adopted to measure and evaluate the surface physical parameters of the fruit, including weight, volume, diameter, and thickness. Let  $x$ ,  $y$ , and  $z$  be the length, width, and thickness of a spheroid fruit, respectively. Then, the geometric mean diameter  $ZJ_A$  of the fruit can be calculated by:

$$ZJ_A = (xyz)^{\frac{1}{3}} \quad (20)$$

The equivalent diameter  $ZJ_E$  and arithmetic diameter  $ZJ_F$  can be respectively calculated by:

$$ZJ_E = \left[ \frac{x(yz)^2}{4} \right]^{\frac{1}{3}} \quad (21)$$

$$ZJ_F = \frac{x + y + z}{3} \quad (22)$$

The sphericity  $\gamma$  of the spheroid fruit can be calculated by:

$$\gamma = \frac{4\pi \left( \frac{3U_o}{4\pi} \right)^{\frac{2}{3}}}{BD} \quad (23)$$

The surface area  $BD$  can be calculated by:

$$BD = \pi (xyz)^{\frac{2}{3}} \quad (24)$$

The elliptical volume  $U_o$  of the measured fruit can be calculated by:

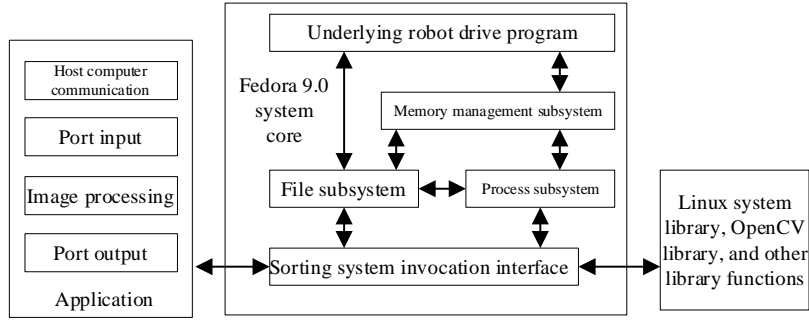
$$U_o = \frac{(xyz)^{\frac{1}{3}}}{x} \quad (25)$$

Table 1. Mean of each physical parameter of the target fruit

Attribute	Fruit maturity		
	Unmature	Mature	Overmature
$x$	48.93	53.1	57.26
$y$	38.125	43.761	38.21
$z$	37.58	54.26	39.38
$ZJ_E$	42.62	45.37	45.35
$ZJ_F$	41.21	44.11	43.52
$ZJ_A$	40.31	43.65	42.76
$\gamma$	40.76	43.81	42.93
$U_o$	37422.8	47567.3	41235.6
$BD$	5621.9	6152.2	5813.8

Table 1 lists the mean of each physical parameter of the target fruit. The shape can largely differentiate between mature, overmature, and unmature fruits. To estimate fruit volume accurately from the pixel length, width, and area in fruit images, this paper collects 2D fruit images in the JPG format, and of the size  $1,600 \times 1,200$ . Let  $H(i, j)$  be a pixel in a collected image, where  $i$  and  $j$  are the abscissa and ordinate, respectively;  $N$  and  $M$  be the length and width of the image, respectively. In shape recognition, the pixel area  $EA$  occupied by the fruit, i.e., the number of image pixels, can be easily obtained:

$$EA = \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} H(i, j) \quad (26)$$



**Figure 6.** Functional modules of image processor

Figure 6 shows the functional modules of image processor. Based on Fedora 9.0 operating system, the image processor calls physical devices via Linux system library, OpenCV library, and other library functions. In the computer vision library OpenCV, the pixels are read one after another column by column from the upper left corner. Once a white pixel is read, it is believed that the tip of the fruit is detected. The length  $x$  and height  $y$  of each column of pixels can be obtained by subtracting the top white pixel from the bottom white pixel:

$$\begin{cases} x = \frac{1}{EA} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} 2iH(i, j) \\ y = \frac{1}{EA} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} 2jH(i, j) \end{cases} \quad (27)$$

Suppose the radius  $v$  of the fruit on each profile equals half the length of the white pixel area. Let  $i$  be the  $i$ -th pixel from the left. Then, the area of all fruit profile can be computed by:

$$\begin{aligned} s[i] &= \pi r^2 \\ EA[i] &= \pi \cdot v^2 \end{aligned} \quad (28)$$

Let  $m$  be the width of fruit pixel. Then, the total volume of a fruit can be derived from the total area of the fruit profiles:

$$\begin{aligned} V_T &= \sum_1^n s[i] \\ U_\phi &= \sum_1^m EA[i] \end{aligned} \quad (29)$$

If the target fruit is not a standard sphere, there will be a large calculated error in the volume derived from the upper, lower, left, and right diameters, which are obtained through image processing. To solve the problem, it is necessary to compute the elliptical factor  $RO$  of the fruit. Let  $RO'$  be the calculated value of  $RO$ ;  $Max(x, y)$  be the maximum horizontal and vertical distance of the ellipsoid. Then, the final roundness  $RO$  can be calculated by:

$$\begin{cases} RO' = \frac{4EA}{\pi [\max(x, y)]^2} \\ RO = \min(RO', 1) \end{cases} \quad (30)$$

Let  $V_T$  be the fruit volume calculated by pixels. Then, the ellipsoid fruit volume can be estimated by:

$$U = RO \cdot U_\phi \quad (31)$$

The color of the fruit also reflects its maturity to a certain extent. In this paper, the color features are extracted from the target fruit by combining the geometric derivation for converting RGB color space to HSI color space with the standard model method. The geometric derivation first extracts the value of the image. In the 2D plane of the dimensionally reduced image, the hue component  $F_H$  of the HSI color space can be calculated based on the vector dot product formula:

$$F_H = \begin{cases} \Phi, & \text{if } F_B \leq F_G \\ 360 - \Phi, & \text{if } F_B > F_G \end{cases} \quad (32)$$

where,  $\Phi$  can be calculated by:

$$\Phi = \arccos \left\{ \frac{\frac{1}{2} [(F_R - F_G) - (F_R - F_B)]}{\sqrt{(F_R - F_G)^2 + (F_R - F_B)(F_G - F_B)}} \right\} \quad (33)$$

The saturation component  $F_S$  can be calculated by:

$$F_S = 1 - \frac{3}{(F_R + F_G + F_B)} [\min(F_R, F_G, F_B)] \quad (34)$$

The value component  $F_I$  can be calculated by:

$$F_I = \frac{1}{3} (F_R + F_G + F_B) \quad (35)$$

Formulas (33)-(35) were imported to OpenCV to compile a program. By running the program, the HSI color space can be drawn, and the three components of the space can be plotted separately. Among them, the hue component map reflects the fruit contour, the saturation component map boasts a strong discriminability, and the value component map presents the color features very clearly.

The standard model method defines the hue of the three basic colors (red, green, and blue) as  $0^\circ$ ,  $120^\circ$ , and  $240^\circ$ , respectively. Suppose  $F_{\max} = \max(F_R, F_G, F_B)$ , and  $F_{\min} = \min(F_R, F_G, F_B)$ . The  $F_H$  component can be calculated by:

$$F_H = \begin{cases} \frac{\pi}{3} \times \frac{F_G - F_B}{F_{\max} - F_{\min}}, & \text{if } F_{\max} = F_R \\ \frac{\pi}{3} \times \frac{F_B - F_R}{F_{\max} - F_{\min}} + \frac{2\pi}{3}, & \text{if } F_{\max} = F_G \\ \frac{\pi}{3} \times \frac{F_R - F_G}{F_{\max} - F_{\min}} + \frac{4\pi}{3}, & \text{if } F_{\max} = F_B \end{cases} \quad (36)$$

If  $F_H < 0$ , then:

$$F_H = F_H + 2\pi, \text{ if } F_H < 0 \quad (37)$$

The corresponding saturation component  $F_S$  and value component  $F_I$  can be respectively calculated by:

$$F_S = F_{\max} = F_{\min} \quad (38)$$

$$F_I = \frac{F_{\max} + F_{\min}}{2} \quad (39)$$

The final step is to extract the eigenvalues from the fruit image. Firstly, each collected image was transformed from the time domain to the frequency domain. Then, the sub bands of the frequency domain were segmented into  $y \times y$  blocks. Each block was assigned a number. Let  $\{\}$  be the operation to solve the nearest integer;  $\Omega(a)$  be the operation to solve the number of elements. Based on the solved integer, a high-dimensional vector  $A_j$  could be solved, which encompasses the column terms of the  $j$ -th frequency domain sub band block  $C_j$ . Then, the vector was divided into  $\omega$  groups. One of the groups can be described by:

$$x = \left\{ \frac{\Omega(a)}{\omega} \right\} \quad (40)$$

The remainder  $t$  can be calculated by:

$$t = \Omega(a) - \omega \times x \quad (41)$$

By method of mean value, the feature points were calculated for the fruit image. Let  $B_i(l)$  be the  $l$ -th element of  $B_i$ ;  $L$  be the

total number of maximum elements among the column terms in each group. Then, the mean value of feature points can be calculated by:

$$\lambda = \frac{1}{L} \sum_{l=1}^L b_i(l) \quad (42)$$

The feature point matrix of the fruit image can be obtained as:

$$CH = \begin{pmatrix} \lambda_{11} & \cdots & \lambda_{1q} \\ \vdots & \ddots & \vdots \\ \lambda_{d1} & \cdots & \lambda_{dq} \end{pmatrix} \quad (43)$$

#### 4. EXPERIMENTS AND RESULTS ANALYSIS

Before extracting the maturity eigenvalues, it is important to segment the original image effectively, and to verify the reasonability of the selected threshold through rigorous experiments. If the threshold is too small, the fruit contour will not be fully displayed; If the threshold is too large, the fruit contour will be so smooth as to distort the image. Since mature pears are between white and yellow, this paper sets the threshold within  $[120, 250]$ . Figure 7 presents the contour extraction effect of pear fruit with the threshold of 210. The segmented fruit contour was rather complete.

Table 2 compares the fruit contours extracted with 14 different thresholds, including  $[120, 250]$ . The fruit contours obtained with the threshold of  $[210, 220]$  were relatively clear and close to the actual situation. Those obtained with the threshold of  $[120, 210]$  were incomplete. Those obtained with the threshold of  $[220, 250]$  were too smooth to meet the requirements.

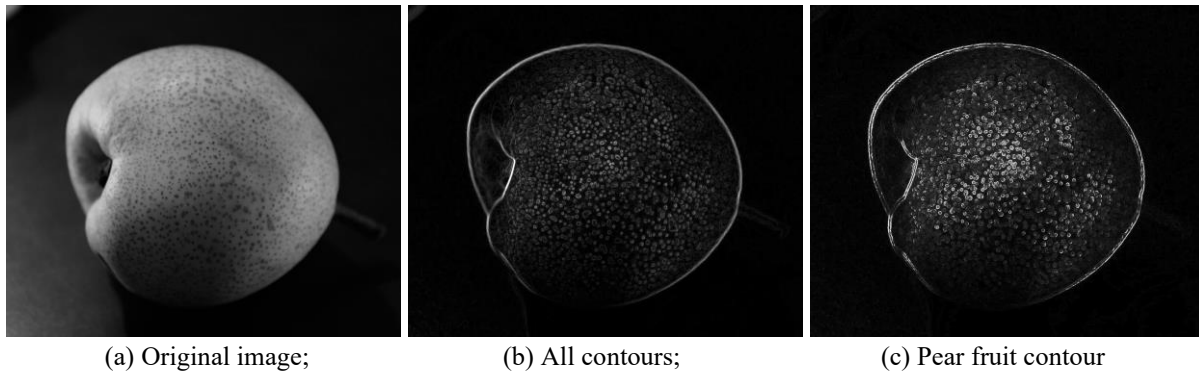


Figure 7. Contour extraction effect of pear fruit

Table 2. Fruit contours extracted with different thresholds

Threshold	All contours	External contours	All contours	External contours	All contours
120	Incomplete	Incomplete	190	Basically complete	Basically complete
130	Incomplete	Incomplete	200	Basically complete	Basically complete
140	Incomplete	Incomplete	210	Complete	Complete
150	Incomplete	Incomplete	220	Clear	Clear
160	Incomplete	Incomplete	230	Smooth	Smooth
170	Incomplete	Incomplete	240	Smooth	Smooth
180	Basically complete	Basically complete	250	Smooth	Smooth

**Table 3.** Recognition effect of fruit volume and maturity

Attribute	Maturity			Recognition accuracy
	Unmature	Mature	Overmature	
Volume	97.24	94.15	91.87	93.46
Maturity	97.48	91.68	96.14	94.51

To verify the effectiveness of the maturity-based robot sorting strategy, several experiments were carried out to recognize the volume and maturity of pears. The mean values of the physical parameters of 100 pears were taken as the sample data to be processed. Another 40 pears were selected for manual recognition. Table 3 presents the recognition effect of fruit volume and maturity. It can be seen that the proposed sorting strategy can effectively measure the maturity of each fruit. The accuracy was as high as 0.9451. The volume estimation was also very accurate, with an accuracy up to 0.9346. The recognition effect meets the basic requirements for sorting.

The sorting system was optimized by our strategy. Then, the distance of the robot from the fruit center was measured, as it grabbed the fruit at 20 different moving speeds. The errors of lagged grabbing and advanced grabbing were summarized. In lagged grabbing, the grabbing position is behind the fruit center; In advanced grabbing, the grabbing position is before the fruit center. The statistical results show that, lagged grabbing occurred 20 times at the moving speed of 8cm/s, 15 at 9.5cm/s, and 8 at 11cm/s. Among them, the mean error was 0.108mm at 8cm/s, 0.055mm at 9.5cm/s, and 0.047mm at 11cm/s.

**Table 4.** Fruit poses in base coordinate system obtained by matching

Fruit number	$v$	$u$	$q$	$\beta$	$\psi$	$\omega$
1	703.61	-32.08	182.64	174.28	3.52	-147.84
2	724.98	-65.12	143.80	176.37	3.45	152.37
3	654.35	-66.72	135.95	174.85	-2.36	-174.29
4	742.75	34.17	141.32	0.45	1.28	-178.45
5	674.62	8.35	136.74	176.72	-1.38	-176.35
6	608.72	46.94	135.86	174.86	1.62	152.45
7	715.34	124.67	132.41	175.31	47.25	176.92
8	842.61	-18.25	157.32	-156.12	36.42	-108.37

**Table 5.** Actual poses of the fruits in the base coordinate system

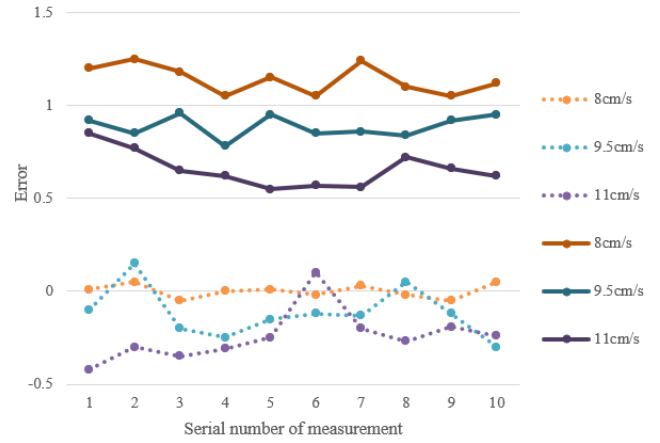
Fruit number	$v$	$u$	$q$	$\beta$	$\psi$	$\omega$
1	708.31	-38.49	185.21	187.56	0.12	147.53
2	742.23	-65.21	135.25	137.28	0.76	-173.21
3	657.69	-71.44	138.76	138.05	0.35	-176.34
4	751.08	31.76	137.94	139.97	-0.37	-177.89
5	675.57	4.25	138.67	136.63	0.48	-176.03
6	615.84	45.63	135.22	177.32	-0.65	148.23
7	714.75	122.32	140.31	176.84	45.31	175.75
8	842.68	-23.75	165.28	-157.65	35.02	-115.40

**Table 6.** Experimental errors

	$v$	$u$	$q$	$\beta$	$\psi$	$\omega$
Mean errors	4.25	3.76	3.52	2.94	2.76	3.72

In the base coordinate system of the robot, the poses of pears in the eight groups of fruit piles were counted and sorted out. In Table 4, the pose information of each pear is obtained from the hand-and-eye calibration results of the robot, and the fine registration results in the preceding section. Note that  $v$ ,  $u$ , and  $q$  represent the centroid position of the fruit surface point cloud

Figure 8 compares the effects before and after grabbing displacement is compensated by our strategy. The dotted line and solid line respectively stand for the compensated displacement and original displacement at each moving speed before and after the compensation, respectively. The compensation error was basically controlled within 0.2mm. However, advanced grabbing occurred at the moving speeds of 9.5cm/s and 11cm/s. Such an error is inevitable, because the advanced grabbing mainly comes from grabbing disturbances.

**Figure 8.** Effects before and after compensating for grabbing displacement

in the base coordinate system;  $\beta$ ,  $\psi$  and  $\omega$  represent the rotation angles of the target about axes  $v$ ,  $u$ , and  $q$ , respectively.

Based on the poses in Table 4, this paper completes the program control of the moving and grabbing of the robot in RobotStudio. During the fruit sorting experiment, the robot firstly grabbed fruits according to the fruit recognition results and pile height, that is, grabbed the fruits on the top of the pile. Next, the targets with two identifiable angles were grabbed earlier than those with three identifiable angles, following the sequence of fine registration. The actual poses of the fruits in the base coordinate system can be read from the demonstrator

of Robot Studio (Table 5).

Comparing Tables 4 and 5, the rotation angle of any fruit rotating about one of the three axes (v, u, and q) changed in  $(-180^\circ, 180^\circ]$ . Table 6 presents the errors between the fruit poses obtained in Robot Studio and the actual poses in the sorting scene.

As shown in Table 6, the mean errors on axes v, u, and q were 4.25mm, 3.76mm, and 3.52mm, respectively. The total mean displacement error stood at 3.84mm. The errors of the rotation angles ( $\beta$ ,  $\psi$  and  $\omega$ ) about the axes v, u, and q were  $2.94^\circ$ ,  $2.73^\circ$ , and  $3.43^\circ$ , respectively. The mean angular error stood at  $3.03^\circ$ . Through the sorting experiments on various kinds of fruits, the proposed 3D point cloud matching algorithm is effective in fruit recognition, and the proposed maturity-based sorting strategy is also feasible. Our strategy can sort different kinds of randomly piled fruits, and complete unordered sorting with centimeter accuracy.

## 5. CONCLUSIONS

This paper studies the target positioning and sorting strategy of fruit sorting robot based on image processing. The authors constructed a visual sorting system for fruit sorting robot, and explained the way to recognize objects in 3D scene and to reconstruct the spatial model based on sorting robot. Next, the maturity of the identified fruits was considered the prerequisite of dynamic sorting of fruit sorting robot, followed by the design of the program flow of the fruit sorting robot. The pear contour extraction effect of our strategy was obtained through experiments. The results show that the fruit contours were relatively complete at the threshold of 210. In addition, more experiments were carried out to verify the fruit volume and maturity recognition effect, and count the errors between the fruit poses obtained in Robot Studio and the actual poses in the sorting scene. The results confirm that the proposed 3D point cloud matching algorithm is effective in fruit recognition, and the proposed maturity-based sorting strategy is also feasible. Our strategy can sort different kinds of randomly piled fruits, and complete unordered sorting with centimeter accuracy.

## ACKNOWLEDGMENT

This work is supported by Key Research and Development Program, Shaanxi Province, China (Grant No.: 2019NY-171, 2019ZDLNY02-04), National Natural Science Foundation of China (Grant No.: 31971805), and National Key Research and Development Program of China (Grant No.: 2019YFD1002401). The authors are also grateful to the reviewers for their insightful comments and suggestions, which helps to make the presentation of this manuscript better.

## REFERENCES

[1] Kumari, N., Bhatt, A.K., Dwivedi, R.K., Belwal, R. (2021). Hybridized approach of image segmentation in classification of fruit mango using BPNN and discriminant analyzer. *Multimedia Tools and Applications*, 80(4): 4943-4973. <https://doi.org/10.1007/s11042-020-09747-z>

[2] Ponce, J.M., Aquino, A., Andújar, J.M. (2019). Olive-fruit variety classification by means of image processing

and convolutional neural networks. *IEEE Access*, 7: 147629-147641. <https://doi.org/10.1109/ACCESS.2019.2947160>

[3] Septiari, A., Hamdani, H., Hatta, H.R., Kasim, A.A. (2019). Image-based processing for ripeness classification of oil palm fruit. In *2019 5th International Conference on Science in Information Technology (ICSITech)*, pp. 23-26. <https://doi.org/10.1109/ICSITech46713.2019.8987575>

[4] Chithra, P.L., Henila, M. (2021). Apple fruit sorting using novel thresholding and area calculation algorithms. *Soft Computing*, 25(1): 431-445. <https://doi.org/10.1007/s00500-020-05158-2>

[5] Sihombing, P., Tommy, F., Sembiring, S., Silitonga, N. (2019). The citrus fruit sorting device automatically based on color method by using tcs320 color sensor and arduino uno microcontroller. In *Journal of Physics: Conference Series*, 1235(1): 012064. <https://doi.org/10.1088/1742-6596/1235/1/012064>

[6] Sidehabi, S.W., Suyuti, A., Areni, I.S., Nurtanio, I. (2018). The Development of Machine Vision System for Sorting Passion Fruit using Multi-Class Support Vector Machine. *Journal of Engineering Science & Technology Review*, 11(5): 178-184. <https://doi.org/10.25103/jestr.115.23>

[7] Khoje, S. (2018). Appearance and characterization of fruit image textures for quality sorting using wavelet transform and genetic algorithms. *Journal of Texture Studies*, 49(1): 65-83. <https://doi.org/10.1111/jtxs.12284>

[8] Wahyuni, S.N., Affan, R. (2020). Oil Palm Sorting System of Fresh Fruit Bunch (FFB) Using Forward Chaining Algorithm. In *Journal of Physics: Conference Series*, 1501(1): 012018. <https://doi.org/10.1088/1742-6596/1501/1/012018>

[9] Henila, M., Chithra, P. (2020). Segmentation using fuzzy cluster-based thresholding method for apple fruit sorting. *IET Image Processing*, 14(16): 4178-4187. <https://doi.org/10.1049/iet-ipr.2020.0705>

[10] Gill, J., Girdhar, A., Singh, T. (2019). Enhancement-based background separation techniques for fruit grading and sorting. *International Journal of Intelligent Systems Technologies and Applications*, 18(3): 223-256. <https://doi.org/10.1504/IJISTA.2019.099342>

[11] Tsuta, M., Yoshimura, M., Kasai, S., Matsubara, K., Wada, Y., Ikehata, A. (2019). Prediction of internal flesh browning of "Fuji" apple using Visible-Near Infrared spectra acquired by a fruit sorting machine. *Japan Journal of Food Engineering*, 20(1): 7-14.

[12] Jiang, H., Wang, B., Chu, X., Zhao, X. (2018). Design and experimental research of shelling and sorting machine for fresh *Camellia oleifera* fruit based on physical properties. In *2018 ASABE Annual International Meeting*, 1. <https://doi.org/10.13031/aim.201800815>

[13] Apte, S.K., Patavardhan, P.P. (2021). Feature fusion based orange and banana fruit quality analysis with textural image processing. In *Journal of Physics: Conference Series*, 1911(1): 012023. <https://doi.org/10.1088/1742-6596/1911/1/012023>

[14] Behera, S.K., Sethy, P.K., Sahoo, S.K., Panigrahi, S., Rajpoot, S.C. (2021). On-tree fruit monitoring system using IoT and image analysis. *Concurrent Engineering*, 29(1): 6-15. <https://doi.org/10.1177/1063293X20988395>

[15] Ding, G., Qiao, Y., Yi, W., Fang, W., Du, L. (2021). Fruit



- fly optimization algorithm based on a novel fluctuation model and its application in band selection for hyperspectral image. *Journal of Ambient Intelligence and Humanized Computing*, 12(1): 1517-1539. <https://doi.org/10.1007/s12652-020-02226-1>
- [16] Syaifuddin, A., Mualifah, L.N.A., Hidayat, L., Abadi, A.M. (2020). Detection of palm fruit maturity level in the grading process through image recognition and fuzzy inference system to improve quality and productivity of crude palm oil (CPO). In *Journal of Physics: Conference Series*, 1581(1): 012003. <https://doi.org/10.1088/1742-6596/1581/1/012003>
- [17] Devi, P.K. (2020). Image segmentation K-means clustering algorithm for fruit disease detection image processing. In *2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, pp. 861-865. <https://doi.org/10.1109/ICECA49313.2020.9297462>
- [18] Chai, R. (2021). Otsu's image segmentation algorithm with memory-based fruit fly optimization algorithm. *Complexity*, 2021: Article ID 5564690. <https://doi.org/10.1155/2021/5564690>
- [19] Siddiqi, R. (2020). Comparative performance of various deep learning based models in fruit image classification. In *Proceedings of the 11th International Conference on Advances in Information Technology*, pp. 1-9. <https://doi.org/10.1145/3406601.3406619>
- [20] Raka, S., Kamat, A., Chavan, S., Tyagi, A., Soygaonkar, P. (2019). Taste-wise fruit sorting system using thermal image processing. In *2019 IEEE Pune Section International Conference (PuneCon)*, pp. 1-4. <https://doi.org/10.1109/PuneCon46936.2019.9105726>
- [21] Ayyub, S.R.N.M., Manjramkar, A. (2019). Fruit disease classification and identification using image processing. In *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, pp. 754-758. <https://doi.org/10.1109/ICCMC.2019.8819789>
- [22] Veni, M., Meyyappan, T. (2019). Digital image watermark embedding and extraction using oppositional fruit Fly algorithm. *Multimedia Tools and Applications*, 78(19): 27491-27510. <https://doi.org/10.1007/s11042-019-7650-0>
- [23] Pandey, C., Sethy, P.K., Biswas, P., Behera, S.K., Khan, M.R. (2020). Quality evaluation of pomegranate fruit using image processing techniques. In *2020 International Conference on Communication and Signal Processing (ICCSP)*, pp. 0038-0040. <https://doi.org/10.1109/ICCSP48568.2020.9182232>
- [24] Ramya, R., Kumar, P., Sivanandam, K., Babykala, M. (2020). Detection and classification of fruit diseases using image processing & cloud computing. In *2020 International Conference on Computer Communication and Informatics (ICCCI)*, pp. 1-6. <https://doi.org/10.1109/ICCCI48352.2020.9104139>
- [25] Liu, J., Tan, J., Qin, J., Xiang, X. (2020). Smoke image recognition method based on the optimization of SVM parameters with improved fruit fly algorithm. *KSII Transactions on Internet and Information Systems (TIIS)*, 14(8): 3534-3549. <https://doi.org/10.3837/tiis.2020.08.022>
- [26] Huynh, T., Tran, L., Dao, S. (2020). Real-time size and mass estimation of slender axi-symmetric fruit/vegetable using a single top view image. *Sensors*, 20(18): 5406. <https://doi.org/10.3390/s20185406>