Tracking and recognition algorithm for a robot harvesting oscillating apples

Qinghua Yang*, Chen Chen, Jiayu Dai, Yi Xun, Guanjun Bao

(*Key Laboratory of Special Purpose Equipment and Advanced Processing Technology, Ministry of Education and Zhejiang Province, Zhejiang University of Technology, Hangzhou 310023, China*)

Abstract: Apple fruits on trees tend to swing because of wind or other natural causes, therefore reducing the accuracy of apple picking by robots. To increase the accuracy and to speed up the apple tracking and identifying process, tracking and recognition method combined with an affine transformation was proposed. The method can be divided into three steps. First, the initial image was segmented by Otsu's thresholding method based on the two times Red minus Green minus Blue (2R-G-B) color feature; after improving the binary image, the apples were recognized with a local parameter adaptive Hough circle transformation method, thus improving the accuracy of recognition and avoiding the long, time-consuming process and excessive fitted circles in traditional Hough circle transformation. The process and results were verified experimentally. Second, the Shi-Tomasi corners detected and extracted from the first frame image were tracked, and the corners with large positive and negative optical flow errors were removed. The affine transformation matrix between the two frames was calculated based on the Random Sampling Consistency algorithm (RANSAC) to correct the scale of the template image and predict the apple positions. Third, the best positions of the target apples within 1.2 times of the prediction area were searched with a de-mean normalized cross-correlation template matching algorithm. The test results showed that the running time of each frame was 25 ms and 130 ms and the tracking error was more than 8% and 20% in the absence of template correction and apple position prediction, respectively. In comparison, the running time of our algorithm was 25 ms, and the tracking error was less than 4%. Therefore, test results indicate that speed and efficiency can be greatly improved by using our method, and this strategy can also provide a reference for tracking and recognizing other oscillatory fruits.

Keywords: apple picking robot, tracking and recognition algorithm, oscillating apple, Hough transform, pyramid LK optical flow algorithm, affine transform, template matching

DOI: 10.25165/j.ijabe.20201305.5520

Citation: Yang Q H, Chen C, Dai J Y, Xun Y, Bao G J. Tracking and recognition algorithm for a robot harvesting oscillating apples. Int J Agric & Biol Eng, 2020; 13(5): 163–170.

1 Introduction

Because manual picking for large-scale apple harvesting is time-consuming and labor-intensive, scientists are developing apple-picking robots, for which the machine's vision capability is one of the most important aspects. The accuracy and efficiency of apple recognition and tracking are very important for effective implementation. Consequently, there has been a wide range of studies done on fruit recognition by researchers all over the world. Nowadays, the commonly used segmentation and recognition algorithms mainly include the threshold segmentation method[1-6], K-means clustering algorithm[7], Fuzzy C-means algorithm (FCM)[8], region growth method[9], Least Squares SVM algorithm (LS-SVM)[10], Artificial Neural Network (ANN)[11], K-nearest Neighbor method (KNN)[12], or combinations of multiple algorithms[13-16]. Algorithms based on convex shell[17-19], concave point[20,21], Hough transform[22], chain code and other methods, have also been used to identify overlapping apples.

However, the above studies were all about static objects, which are challenging to effectively apply to practical conditions. In actual picking, wind force and the physical picking action can cause apples to swing and change positions. Thus workers need to wait for the apples to be still or to image repeatedly in order to accurately identify and locate apples, significantly reducing the picking efficiency. In view of these problems, Zhao et al.[23] and Lyu et al.[24] used a template matching method to track the apple position based on the information association of the initial and later images, but it was reported the extracted templates could not cope with the changes of image scale and angle during the camera's approach to the apple, resulting in large errors in tracking. Feng et al.[25] proposed a SURF feature and the RANSAC model for solving affine transformation parameters obtained the target scale information and used a compression tracking algorithm to track target fruit. Because the algorithm needed to use a compression feature to describe target fruit, it was difficult to ensure the real-time efficiency and performance of the algorithm.

In this study, a novel tracking and recognition method for an apple-picking robot was developed. A camera mounted on the execution end of the system was used to obtain an image and used local parameter adaptive Hough transform to identify the apple location in the initial image. In the follow-up images, the correlation between images was used to track the apple. Firstly, the pyramid Lucas-Kanade (LK) optical flow algorithm was used to track the Shi-Tomasi corner, then the affine transformation matrix between the earlier and later frames was calculated, and the base image was used to correct the template and predict the approximate position of the apple. Finally, the template matching was used around the prediction area to track the position of the apple.

2 Apple recognition

Due to limited experimental conditions and orchard availability,

Received date: 2019-11-10 **Accepted date:** 2020-07-15

Biographies: Chen Chen, Master candidate, research interest: machine vision, Email: 724435042@qq.com; **Jiayu Dai,** Master candidate, research interest: machine vision, Email: 1091793708@qq.com; **Yi Xun,** PhD, Lecturer, research interest: machine vision, Email: xunyi@zjut.edu.cn; **Guanjun Bao,** PhD, Professor, research interest: soft robotics, Email: gjbao@zjut.edu.cn.

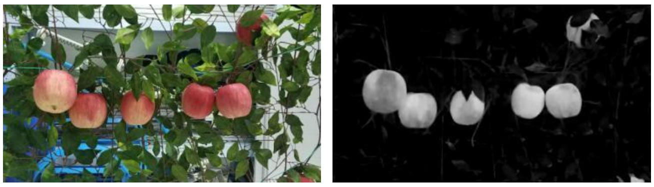
***Corresponding author: Qinghua Yang,** PhD, Professor, research interest: agriculture robot. Zhejiang University of Technology, Hangzhou 310023, China. Tel: +86-15397101399, Email: zjutme@163.com.

in this study, an apple tree model which consisted of dwarf planting was used to simulate trees in a laboratory environment. For this apply picking process, it is necessary for the robot to accurately recognize the images first.

2.1 Image preprocessing

There is noise in the images collected by the image acquisition equipment. In order to reduce the noise and preserve the edges of the apples as much as possible, median filtering was used on the image.

Due to the red color of the mature apples, they are distinct colorwise from other objects in the image. In order to improve the distinction between apples and background, three-channel RGB images were transformed into single-channel gray images, which are more convenient for the subsequent application of algorithms. The two times Red minus Green minus Blue (2R-G-B) color feature was used to process the original images, and a typical result is shown in Figure 1.



a. Before preprocessing b. After preprocessing

Figure 1 Images before and after preprocessing of median filtering and (2R-G-B) color transformation

2.2 Image segmentation

Threshold-based segmentation is a common method to segment objects, apples, and their background. After preprocessing, the gray level between the apple and background was dramatically different, as demonstrated by the histogram analysis of the image, which shows obvious peaks and troughs (Figure 2).

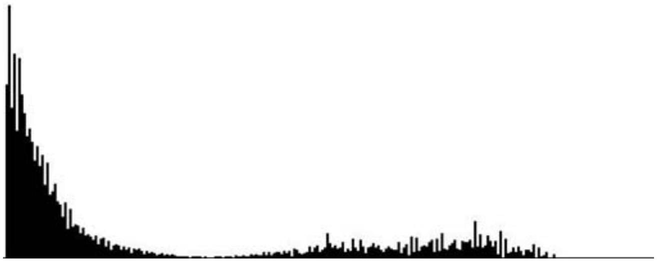


Figure 2 Histogram of (2R-G-B) image

In order to improve the robustness of illumination, Otsu's[26] method was used to segment the threshold. Otsu's method is an algorithm to obtain the best threshold value for the image histogram by iterating the threshold value until the intra-class variance becomes the minimum and the inter-class variance becomes the maximum. The effect is shown in Figure 3.

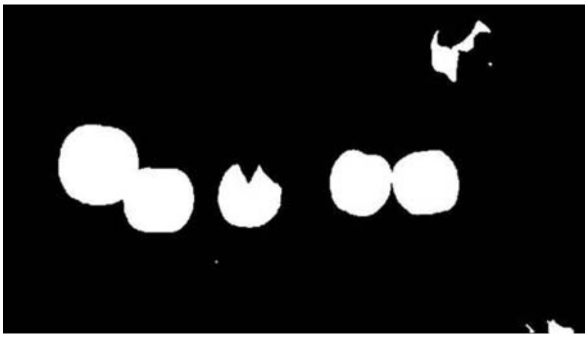


Figure 3 Segmentation of apple image by Otsu's Method

2.3 Image perfection

There were many defects in the binary images of the apples, which would cause issues for apple recognition. In this study, a hollow filling algorithm was used to fill the space formed by the calyx at the tail of apple fruit. Open operations were performed on the image using a circular template with a radius of 3 pixels to remove little noise. After these processing steps, the binary image might still retain some areas that have no apples but rather have other objects like branches and leaves, which are shown as non-circular irregular areas in the image. For this reason, the minimum length to width ratio for the bounding rectangle of the connected region was defined as L, and the connected regions with L>2.5 were removed. The following equations were applied:

$$L=\begin{cases} W/H & (W>H) \\ H/W & (W<H) \end{cases} \tag{1}$$

where, W is the minimum bounding rectangle width; H is the minimum bounding rectangle height. At the same time, there may be apples that remain far away and show up as a small circular connected region. Therefore, S_T is defined as the area threshold and the connected regions with area $S<S_T$ were removed.

$$S_T=\frac{S_{\max}}{2*L} \tag{2}$$

where, S_{max} is the area of the largest connected area.

In order to prevent the apple tree or the machine body from being damaged when picking apples that are seriously blocked by the branches and leaves, F was defined as the filling rate of the connected area, and the connected areas with F<0.5 were removed.

$$F=\frac{S_0}{S_t} \tag{3}$$

where, S_0 is the area of the connected area; and S_T is the minimum area of the rectangle surrounded by the connected area.

The improved binary image is shown in Figure 4.

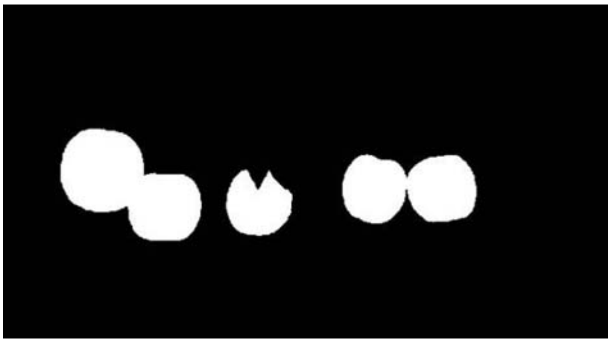


Figure 4 Binary images after processing of hollow filling, open operation and error area removal

2.4 Apple recognition

According to reference [27], because of different camera angles, differences in the natural environment, varied apple shapes, and apples with other defects such as occlusion overlap. The Hough circle transformation algorithm was proposed to fit and extract apple shapes.

However, depending on the distance and placement of the camera, the sizes of the apples in an image might be significantly different, making it challenging to provide accurate parameters for the Hough circle transformation algorithm. Also, this could easily result in too many output fitting circles, leading to algorithm failure, and it could take a long time to perform the global Hough transform, consequently having poor real-time performance.

In order to solve the above problems, a local parameter adaptive Hough transform method was proposed. The algorithm was as follows:

Step 1: The Hough gradient method was used to find the circle in the rectangular range for each connected region of the image. If the radius R=the width of the smallest bounding rectangle in the connected region×0.5, the parameters of the Hough transform were: the maximum radius $R_{\max}=R\times1.2$, the minimum radius $R_{\min}=R/1.2$, and the minimum center distance $D=R_{\min}$.

Step 2: The circle whose center pixel is 0, area $S<S_t$, filling ratio $F<0.5$ was removed. The remaining circles were regarded as the correct apple targets for extraction, and the algorithm ended.

The effect of apple recognition using this algorithm is shown in Figure 5.

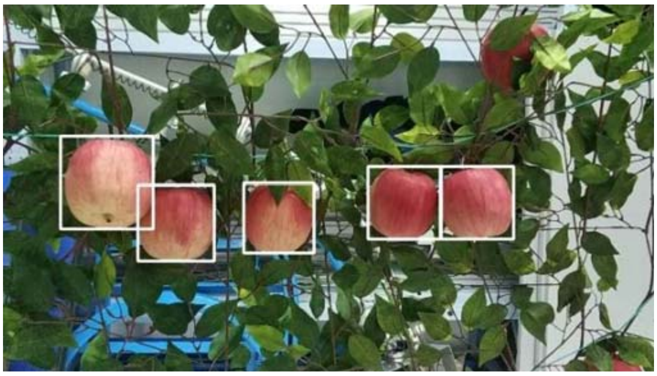
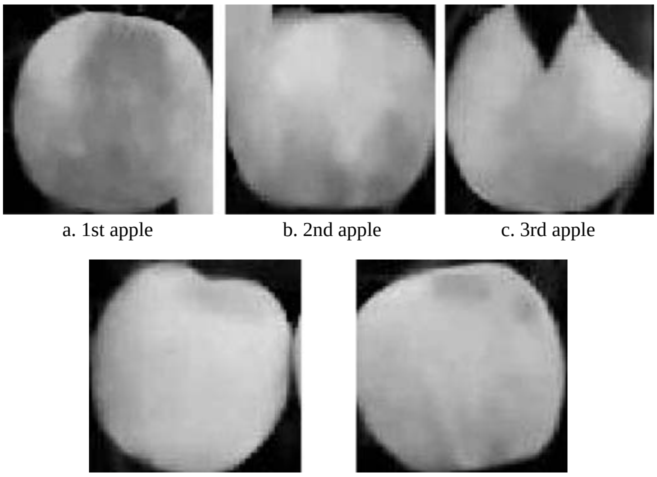


Figure 5 Apple recognition

3 Apple target fast tracking

3.1 Template and corner extraction

In order to track the position of the apple, the rectangular region of the target fruit was extracted as the template image in a 2R-G-B image. The extracted template images for an example apple set are shown in Figures 6a-e from left to right.



a. 1st apple b. 2nd apple c. 3rd apple

d. 4th apple e. 5th apple

Figure 6 Template image

In target tracking, corners are a type of common tracking feature, which has rotation invariance and robustness to illumination. For the images in this paper, the Shi-Tomasi corners[28] were extracted for subsequent tracking, corner points were extracted from a gray-scale image, and the overall extraction effect is shown in Figure 7. Corner points are represented by black dots in the image.

3.2 Corner feature tracking

There is an association of information between the front and back images captured by the system. Using this information association, the target apple can be tracked more quickly and accurately. In this paper, the pyramid LK optical flow method was used to calculate the position of the Shi-Tomasi corner in the subsequent frame, which was used to calculate the affine transformation matrix.

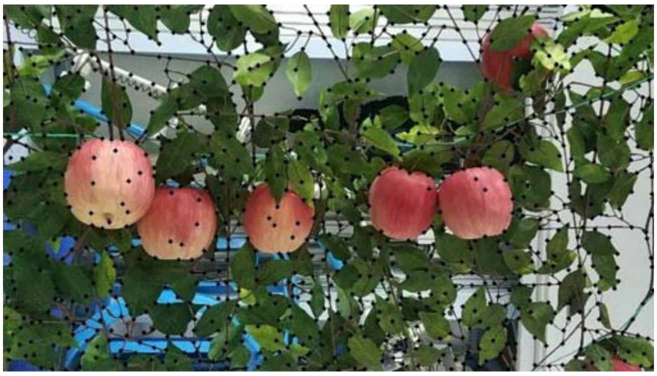


Figure 7 Corner extraction by Shi-Tomasi's method

During the picture moving process, the optical flow is the rate of each pixel moving in the *X*, *Y* direction. An LK optical flow algorithm was implemented based on the following three assumptions:

1. Constant brightness;
2. Continuous time is short or movement is small;
3. The pixels in the same neighborhood have the same motion.

Based on the first two assumptions, the constraint equation of the two-dimensional optical flow could be obtained:

$$\frac{\partial_I}{\partial_x}\delta x+\frac{\partial_I}{\partial_y}\delta_y+\frac{\partial_I}{\partial_t}\delta_t=0 \tag{4}$$

However, the above equation cannot be solved by only one point. Based on the third hypothesis, the least square method was used to solve for all points in the neighborhood of corner points:

$$\begin{bmatrix} I_{x_1}I_{y_1} \\ I_{x_2}I_{y_2} \\ \vdots \ \vdots \\ I_{x_n}I_{y_n} \end{bmatrix}\begin{bmatrix} V_x \\ V_y \end{bmatrix}=\begin{bmatrix} -I_{t_1} \\ -I_{t_2} \\ \vdots \\ -I_{t_n} \end{bmatrix} \tag{5}$$

$$\vec{v}=(A^TA)^{-1}A^T(-b) \tag{6}$$

Since the LK optical flow is based on the above three assumptions, it is only suitable when the time is extremely short or the displacement is extremely small. In order to be able to meet the strict displacement constraint, it is necessary to construct an image pyramid, and continuously solve it from the top of the pyramid to improve the accuracy of optical flow. The result of implementing pyramid LK optical flow for tracking the Shi-Tomasi corner effect is shown in Figure 8. The black lines in the figure are connected to the position of the same corner between the two frames, and the black dots indicate the position of the corners in the current frame.

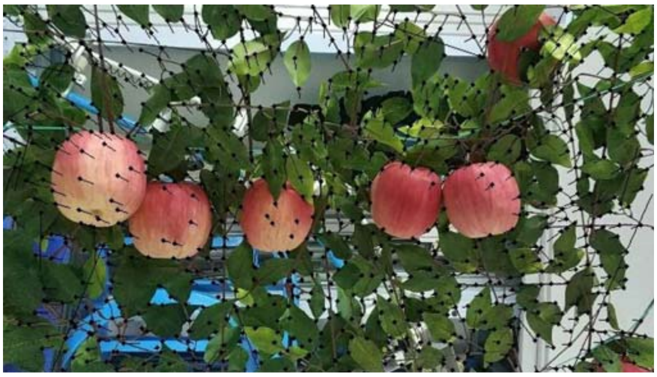


Figure 8 Corner tracking by Pyramid LK Optical Flow

For real-world data, there might be errors in the tracking of pyramid LK optical flow. In order to improve the robustness of tracking, the forward and backward optical flow error method was used to eliminate the unstable corner points. The optical flow method was first used to forwardly track the corner points, and then the optical flow method was used to backtrack to the original image. If a corner point changed more than 0.01 pixels before and after the tracking, the corner point would be considered to be unstable and would be rejected. The picture after eliminating the unstable corner points is shown in Figure 9.

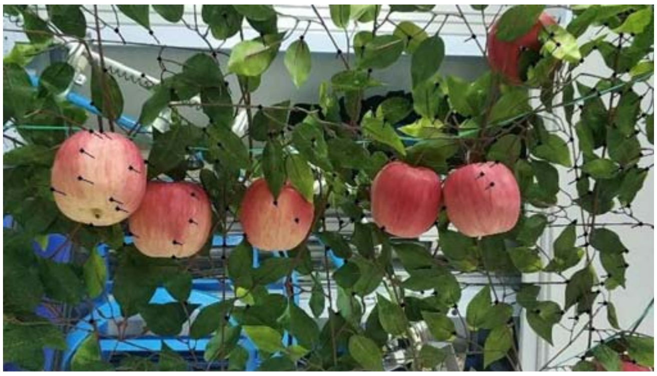


Figure 9 Remove the unstable corners

3.3 Matching recognition

Template matching, the most primitive and basic pattern recognition method, can be used to find the position of the target apple template in the subsequent image. The equation is as follows:

$$R(x,y)=\frac{\sum_{i}^{M}\sum_{j}^{N}(T(i,j)-\overline{T})I(x+i,y+j)-\overline{I})}{\sqrt{\sum_{i}^{M}\sum_{j}^{N}(T(i,j)-\overline{T})^{2}\cdot\sum_{i}^{M}\sum_{j}^{N}(I(x+i,y+j)-\overline{I})^{2}}} \tag{7}$$

where, M and N are the template sizes; T is the template pixel value; I is the image pixel value to be matched; T is the template average pixel value, and I is the average pixel value of the image to be matched.

Template matching has limitations, predominantly because it can only work in parallel. If the matching target in the image rotates or changes in scale, the algorithm is invalid. Because there are many similar apples in the image, if the matching area is not well predicted, it would be easy to match one apple template image with another similar apple, leading to tracking failure.

In this paper, an affine transformation matrix was used to solve the problems in template matching. The matrix form of affine transformation is:

$$M=\begin{bmatrix}\rho\cos\theta & \rho\sin\theta & t_x\\ -\rho\sin\theta & \rho\cos\theta & t_y\\ 0 & 0 & 1\end{bmatrix} \tag{8}$$

where, M is the affine transformation matrix; ρ is the scale factor; θ is the rotation angle; t_x is the x-direction translation, t_y is the y-direction translation.

Generally, the affine transformation matrix can be solved by only three pairs of points, but sometimes there are more than three pairs of points and possible external points. Therefore, this paper solved the problem by using an affine transformation matrix based on the random sampling consistency algorithm (RANSAC).

The scale factor ρ and rotation angle θ can be decomposed from the affine transformation matrix, and they are used to correct the template and solve the scale and rotation problems in the template matching process.

At the same time, the approximate position of the template image in the current frame can be predicted by using the affine transformation matrix, as shown in Equation (9). In this study, based on the apple position predicted by Equation (10), a de-mean normalized cross-correlation algorithm was used to match the template within the range of 1.2 times the size of the template. This not only solved the problem of a possible mismatch but also reduced the scope of template matching and improved the algorithm speed. This size range was found that, 1.2 times of the template, was optimal, but the range could be adjusted according to the intensity of apple oscillation by the following equations.·

$$\begin{bmatrix}x_c\\ y_c\\ 1\end{bmatrix}=\begin{bmatrix}\rho\cos\theta & \rho\sin\theta & t_x\\ -\rho\sin\theta & \rho\cos\theta & t_y\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}x_r\\ y_r\\ 1\end{bmatrix} \tag{9}$$

which is the same as:

$$\begin{aligned}x_c&=\rho\cos\theta\cdot x_r+\rho\sin\theta\cdot y_r+t_x\\ y_c&=-\rho\sin\theta\cdot x_r+\rho\cos\theta\cdot y_r+t_y\end{aligned} \tag{10}$$

where, ($x_{r,}y_y$) is the coordinate of the apple center in the previous frame; ($x_{c,}y_c$) is the coordinate of the apple center in the current frame.

3.4 Algorithm flow

A flow chart for fast tracking algorithm applied to each apple target between the two frames is shown in Figure 10.

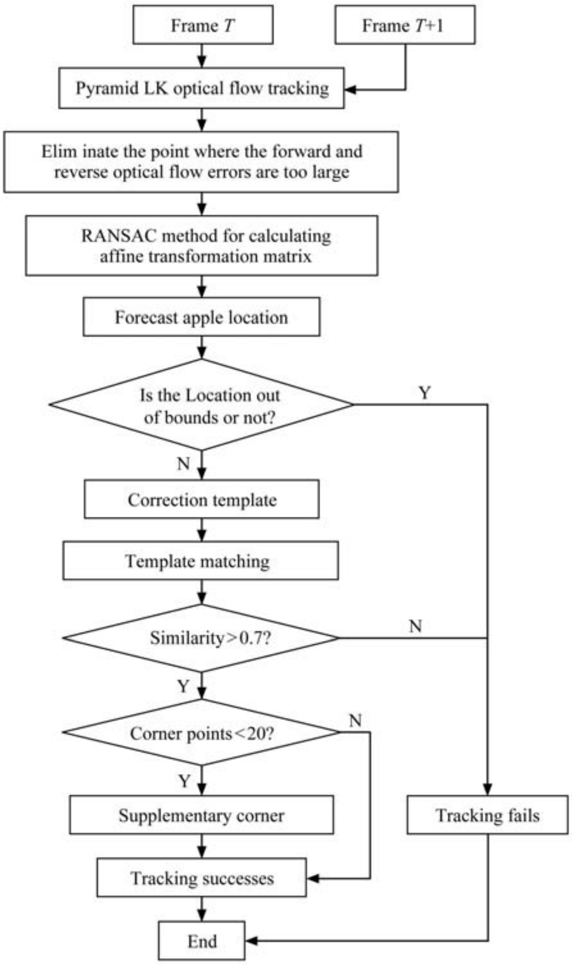

Frame T
Frame T+1
Pyramid LK optical flow tracking
Elim inate the point where the forward and reverse optical flow errors are too large
RANSAC method for calculating affine transformation matrix
Forecast apple location
Is the Location out of bounds or not?
Y
N
Correction template
Template matching
Similarity>0.7?
N
Y
Corner points<20?
N
Y
Supplementary corner
Tracking fails
Tracking successes
End

Figure 10 Apple target fast tracking

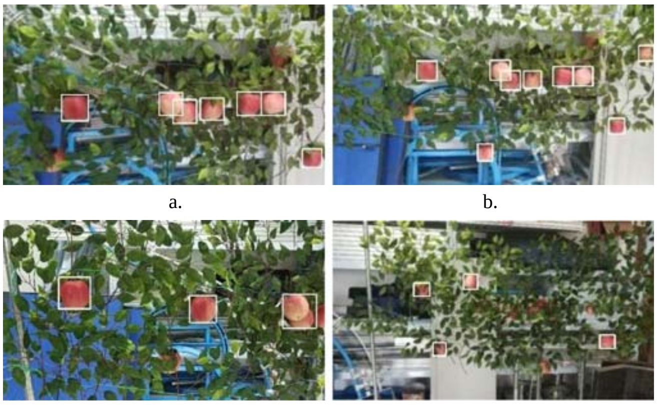
4 Experimental results and analysis

4.1 Experimental condition

The algorithm programming environment was Ubuntu 16.04, and the machine was equipped with an Intel i5-4210M 2.6GHz CPU, 4GB of memory. The OpenCV3.4.3 open source visual library was the main tool used to complete the algorithm programming. The camera was installed on the execution end. In order to improve the processing speed, the 1920×1080-pixel-image collected was compressed to 640×360 size. In the laboratory environment, simulated dwarf apple trees were used.

4.2 Fast recognition of apples

In order to verify the accuracy of apple recognition, various growth postures of apples in the natural environment, such as occlusion and overlap, were simulated. Thirty images of apple trees with different angles and placement were taken. The average running time of the recognition algorithm was 72 ms. The recognition success rate was 91.3% in the laboratory setting, though challenges remain in extrapolating to larger numbers of pictures due to the limited experimental conditions. Representative samples of the recognition results are shown in Figures 11a-11d.



a. b.

c. d.

Figure 11 Representative results for fast recognition algorithm for apples

In most cases, the influence of occlusion and overlap artifacts present in the process of apple recognition can be overcome by the algorithm. Sometimes, as shown in Figure 11c, the three apples close to the border on the right overlap and are not correctly identified by the algorithm.

This was caused by an error in the calculation of the radius parameter range, for the local parameter adaptive Hough transform method. When three or more fruits are overlapping, if their positioning just makes the length-width ratio L of the minimum bounding rectangle close to 1, then the minimum bounding rectangle width will be approximately considered as the apple radius by the algorithm, resulting in recognition errors.

4.3 Fast tracking of apples

In order to evaluate the performance of this tracking algorithm, a number of experiments were carried out. The extracted apple position was taken as the true value of the apple position. The ratio of root mean square error (RMSE) to the diameter of the apple extracted from the first frame was used as the criterion of tracking error.

$$RMSE=\sqrt{\frac{1}{N}\sum_{i=0}^{N-1}[(x_{it}-x_{ir})^2+(y_{it}-y_{ir})^2]/s_i^2} \qquad (11)$$

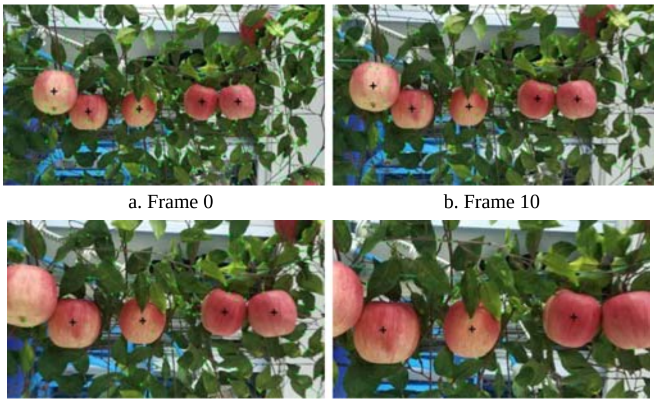
$$Error = RMSE / d \times 100\%$$

where, $RMSE$ is the tracking root mean square error; N is the number of frames; (x_{it}, y_{it}) is the apple centroid coordinate obtained by the tracking algorithm; (x_{ir}, y_{ir}) is the apple centroid coordinate obtained by the recognition algorithm; s_i is the ratio of the first frame to the first frame scale, the error is the tracking error, and d is the apple diameter in the first frame.

4.3.1 Fast tracking of static apples

When there is no wind or other adverse weather, apples are almost motionless. Thus, it is important to examine whether the algorithm is compatible with static apple tracking and recognition.

In a laboratory environment, a set of apples were kept still, and the image sequences were acquired during a process in which the execution end of the camera system gradually approached the apples. The tracking algorithm tracked the static apples as shown in Figures 12a-12d. Black crosses indicate the position of the centers of mass of apples.



a. Frame 0 b. Frame 10

c. Frame 20 d. Frame 30

Figure 12 Representative results for fast tracking algorithm for static apples

The average running time of the tracking algorithm was 25 ms namely 40 frames/s, which is twice the speed of the recognition algorithm. The RMSE, radius d and error of the five apples in the figure are shown in Table 1. The tracking error of each apple was about 2%, which means the results are very close to the apple positions obtained by the recognition algorithm and meet the accuracy requirements of automated apple picking.

Table 1 Static apple tracking error

Apple index	RMSE	d	Error
1	0.73	36	2.03%
2	0.75	64	1.17%
3	0.67	36	1.86%
4	0.76	39	1.95%
5	0.97	46	2.11%

For the static apples, apple positions could be predicted more accurately. After template matching correction, the error could be controlled to be in a very small range.

Error source analysis: This paper used an affine transformation matrix method to correct the template and predict the approximate position of apples. Although the accuracy of the affine transformation matrix has been improved by multiple means in this paper, some small errors, which accumulate with the running of the program, are unavoidable. However, since only a few frames needed to be tracked from the recognition to the final approach to the apple, cumulative errors in the affine transformation matrix had little effect on the template correction. At the same time, having a range of 1.2 times of the area to be searched also resulted in enough redundancy to mitigate the position prediction errors.

4.3.2 Fast tracking of oscillatory apples

Apples tend to swing under the influence of factors such as wind and the picking action. In order to simulate the effect of wind force or picking action, the apple tree was shaken horizontally by hand. The image sequences were captured as the execution end of the camera system approached the apples. Figure 13a-13d demonstrates the successful fast tracking of dynamic apples. Black crosses indicate the positions of the apples' centers of mass.

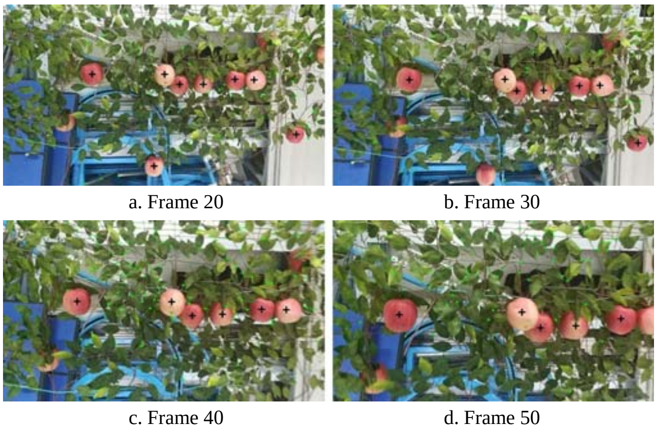

a. Frame 20 b. Frame 30
c. Frame 40 d. Frame 50

Figure 13 Representative results for fast tracking algorithm for dynamic apples

For the process of tracking apples, the running time of the tracking algorithm was still about 25 ms, similar to the static case. The tracking root mean square error (RMSE), radius *d* and error of the 8 apples in the figure are shown in Table 2. Apple numbers were assigned from left to right in the horizontal direction.

Table 2 Concussion apple tracking error

Apple index	RMSE	*d*	Error
1	0.92	34	2.71%
2	0.96	28	3.43%
3	1.26	39	3.23%
4	1.44	43	3.35%
5	1.03	49	2.10%
6	1.40	37	3.78%
7	1.85	41	4.51%
8	1.15	34	3.38%

It is apparent that when used for tracking swinging apples, the system would accumulate tracking errors to a certain extent. The main reason is that on the basis of static apple tracking error, an apple in motion could produce a small angle rotation in its swinging process, thus causing a certain degree of error in template matching. However, because the apple template used in this paper was extracted from a 2R-G-B color difference image, for which the background was almost black, and the apple was round, this made differences in the apple rotation indistinguishable before and after in the 2R-G-B color scheme. As a result, the template would be minimally impaired. However, although this tracking error increases to a certain extent, it remains below 4%, which meets the accuracy requirements for automated apple picking.

4.4 Algorithm comparison experiment

In order to show the superiority of this algorithm relative to other tracking methods, this algorithm was compared with two other algorithms on the same video sequence:

(1) Algorithm 1: the algorithm in this paper.

(2) Algorithm 2[24,25]: in which the template image was not corrected, otherwise the rest was consistent with the algorithm in this paper.

(3) Algorithm 3: in which the prediction for apple position was not carried out otherwise the rest was consistent with the algorithm in this paper.

The operational effectiveness of algorithm 1 is shown in section 4.3.2.

The effectiveness of algorithm 2 for tracking oscillating apples is shown in Figures 14a-14d. Black crosses indicate the positions of the centers of mass of apples.

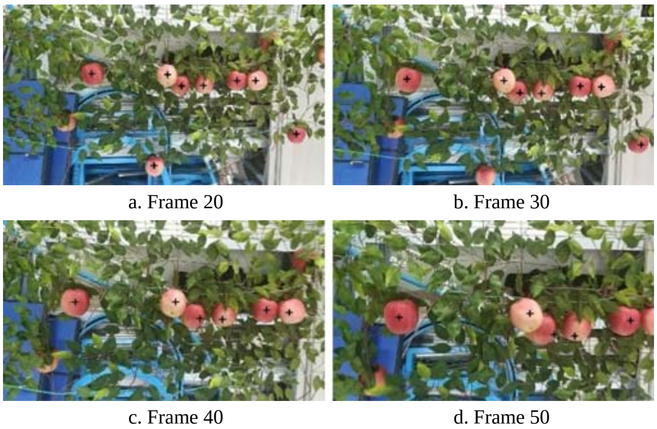

a. Frame 20 b. Frame 30
c. Frame 40 d. Frame 50

Figure 14 Results for algorithm 2 for dynamic apples

The running time of the algorithm was about 25 ms. The tracking error was larger than 8%, as shown in Table 3. The reason for the increase in the tracking error is that as the camera approached the apples, the image scale became larger and larger, while the template image scale was not modified; also there was a large gap between the image scale and the template image scale, directly leading to a large error in template matching.

Table 3 Tracking error using algorithm 2

Apple index	RMSE	*d*	Error
1	3.04	34	8.94%
2	2.40	28	8.57%
3	3.76	39	9.64%
4	10.62	43	24.70%
5	6.66	49	13.60%
6	3.14	37	8.49%
7	4.04	41	9.85%
8	1.19	34	3.50%

The operational effectiveness of algorithm 3 is shown in Figures 15a-15d. Black crosses indicate the positions of the apples' centers of mass.

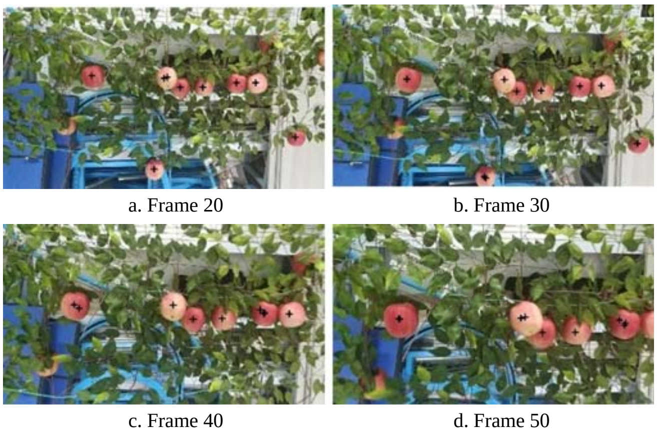

a. Frame 20 b. Frame 30
c. Frame 40 d. Frame 50

Figure 15 Results for algorithm 3 for dynamic apples

Because of lacking any prediction of apple locations, template matching must be carried out in the range of the whole image, resulting in a long-running time (about 130 ms). At the same time, due to the similarity between apples, a few apple templates were matched to other apples, causing matching errors. The tracking errors are presented in Table 4.

Tab 4 Tracking error using algorithm 3

Apple index	RMSE	*d*	Error
1	3.22	34	9.47%
2	6.52	28	23.28%
3	2.28	39	5.84%
4	2.79	43	6.48%
5	2.94	49	6.00%
6	3.50	37	9.46%
7	67.86	41	165.51%
8	18.37	34	54.03%

The comparison of the tracking errors of the three algorithms is shown in Figure 16. It can be seen that the algorithm in this paper has the smallest error and best stability compared with the other two, indicating that it is most suitable for a stable and robust apple picking robot.

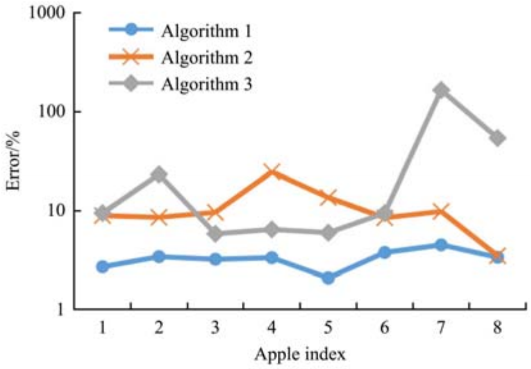


Figure 16 Comparison of tracking errors of three algorithms

5 Conclusions

(1) In this paper, an algorithm for tracking and identifying oscillating apples, with facile implementation into apple-picking robots, was proposed. In this method, the first image was medianly filtered, a 2R-G-B chromatogram was calculated, and then Otsu's method was used for image segmentation. In order to overcome the problem of global Hough transform, a local parameter adaptive Hough transform was used to recognize apple images. Experiments showed that the average running time of the apple recognition algorithm in the laboratory environment was 72 ms, and the recognition success rate was 91.3%.

(2) In order to track target apples, the algorithm extracted the apple template image and the Shi-Tomasi corner of the image from the first frame. In the following image, the pyramid LK optical flow method was used to track the corner points, and the positive and negative optical flow error method was used to remove the corner points with large tracking errors. Then, an affine transformation matrix was calculated based on the random sampling consistency algorithm (RANSAC), which was used to correct the template scale as well as the angle and predict the approximate positions of the apples. Finally, the best positions of the apples were searched in the range of 1.2 times of the predicted apple regions by using the algorithm of de-mean normalized cross-correlation template matching. The experimental results showed that the algorithm could track the apples well regardless of whether they were still or oscillating. The average running time of the algorithm was 25 ms, the tracking error for static apples was less than 2%, and the tracking error for oscillating apples was less than 4%. Compared with other algorithms, the real-time performance and accuracy of the algorithm were greatly improved, thus meeting the requirements for fast and accurate apple picking robots.

(3) At present, there persist some defects in the algorithm that need further improvement. First, when the recognition algorithm is faced with multiple overlapping apples, if the apples are not arranged in a straight line, it is easy to obtain recognition errors. Second, there are accumulating errors in the affine transformation matrix, which is the main source of tracking errors. Third, this paper has not yet tested the robustness of the algorithm in different lighting environments. However, because the pyramid LK optical flow is robust in different lighting conditions, and a variety of methods have been applied to minimize the tracking error, it can be reasonably speculated that the algorithm in this paper will perform well in different lighting conditions.

Acknowledgements

This work was financially supported by Basic Public Welfare Research Project of Zhejiang Province (Grant No. LGN20E050007).

[References]

[1] Jiang G Q, Zhao C J. Apple recognition based on machine vision. 2012 International Conference on Machine Learning and Cybernetics, Xi'an: IEEE, 2012; pp.1148–1151.

[2] Xu Y, Li Y H, Song H B, He D J. Segmentation method of overlapped double apples based on Snake model and corner detectors. Transactions of the CSAE, 2015; 31 (1): 196–203. (in Chinese)

[3] Lyu J D, Zhao D A, Ji W, Ding S H. Recognition of apple fruit in natural environment. Optik, 2016; 127(3): 1354–1362.

[4] Shamshiri R R, Hameed I A., Karkee M, Weltzien C. Robotic harvesting of fruiting vegetables: A simulation approach in V-REP, ROS and MATLAB. In Automation in Agriculture-Securing Food Supplies for Future Generations. IntechOpen, 2018. doi: 10.5772/intechopen.73861.

[5] Shamshiri R R, Weltzien C, Hameed I A., Yule I J, Grift T E, Balasundram S K, et al. Research and development in agricultural robotics: A perspective of digital farming. Int J Agric & Biol Eng, 2018; 11(4): 1–14.

[6] Liu Z D, Lyu Z Q, Zheng W X, Zhang W Z, Cheng X X. Design of obstacle avoidance controller for agricultural tractor based on ROS. Int J Agric & Biol Eng, 2019; 12(6): 58–65.

[7] Wang D D, Song H B, Yu X L, Zhang W Y, Qu W F, Xu Y. An improved contour symmetry axes extraction algorithm and its application in the location of picking points of apples. Spanish Journal of Agricultural Research, 2015; 13(1): e0205. doi: 10.5424/sjar/2015131-6181.

[8] Yang Q H, Chang C, Bao G J, Fan J, Xun Y. Recognition and localization system of the robot for harvesting Hangzhou White Chrysanthemums. Int J Agric & Biol Eng, 2018; 11(1): 88–95.

[9] Ji W, Zhao D A, Cheng F Y, Xu B, Zhang Y, Wang J J. Automatic recognition vision system guided for apple harvesting robot. Computers & Electrical Engineering, 2012; 38(5): 1186–1195.

[10] Yang Q H, Luo S L, Chang C, Xun Y, Bao G J. Segmentation algorithm for Hangzhou white chrysanthemums based on least squares support vector machine. Int J Agric & Biol Eng, 2019; 12(4): 127–134.

[11] Jia W K, Zhao D A, Liu X Y, Tang S P, Ruan C Z, Ji W. Apple recognition based on K-means and GA-RBF-LMS neural network applicated in harvesting robot. Transactions of the CSAE, 2015; 31(18): 175–183. (in Chinese)

[12] Linker R, Cohen O, Naor A. Determination of the number of green apples in RGB images recorded in orchards. Computers and Electronics in Agriculture, 2012; 81: 45–57.

[13] Yang Q H, Jia M M, Xun Y, Bao G J. Detection of egg stains based on local texture feature clustering. Int J Agric & Biol Eng, 2018; 11(1): 199–205.

[14] Gongal A, Silwal A, Amatya S, Karkee M, Zhang Q, Lewis K. Apple crop-load estimation with over-the-row machine vision system. Computers and Electronics in Agriculture, 2016; 120: 26–35.

[15] Nguyen T T, Vandevoored K, Wouters N, Kayacan E, De B J G, Saeys W. Detection of red and bicoloured apples on tree with an RGB-D camera. Biosystems Engineering, 2016; 146: 33–44.

[16] Malik M H, Zhang T, Li H, Zhang M, Shabbir S, Saeed A. Mature tomato fruit detection algorithm based on improved HSV and watershed algorithm. IFAC Papersonline, 2018; 51(17): 431–436.

[17] Song H B, He D J, Pan J P. Recognition and localization methods of occluded based on convex hull theory. Transactions of the CSAE, 2012; 28(22): 174–180. (in Chinese)

[18] Song H B, Zhang C D, Pan J P, Yin X, Zhuang Y B. Segmentation and reconstruction of overlapped apple images based on convex hull. Transactions of the CSAE, 2013; 29(3): 163–168. (in Chinese)

[19] Li L J, Yang H J. Revised detection and localization algorithm for camellia oleifera fruits. Transactions of the CSAM, 2016; 47(12): 285–292, 346. (in Chinese)

[20] Liu W H, Sui Q M. Automatic segmentation of overlapping powder particles based on concavity points. Journal of Electronic Measurement and Instrumentation, 2010; 24(12): 1095–1100. (in Chinese)

[21] Xie Z H, Ji C Y, Guo X Q, Zhu S X. Detection and location algorithm for overlapped fruits based on concave spots searching. Transactions of the CSAM, 2011; 42 (12): 191–196. (in Chinese)

[22] Liu Y K, Gao H L, Zhang J. Improvement of apple picking system design and visual algorithms based on open source project. Jiangsu Agricultural Sciences, 2019; 47(4): 187–192. (in Chinese)

[23] Zhao D A, Shen T, Chen Y, Jia W K. Fast tracking and recognition of overlapping fruit for apple harvesting. Transactions of the CSAE, 2015; 31(2): 22–28. (in Chinese)

[24] Lyu J D, Zhao D A, Ji W. Fast tracking recognition method of target fruit for apple harvesting robot. Transactions of the CSAM, 2014; 45(1): 65–72. (in Chinese)

[25] Feng W, Zhao D A. Fast tracking and recognition of apple harvesting robot for target fruit. Information Technology, 2018; 5: 5–9. (in Chinese)

[26] Otsu N. A Threshold selection method from gray-level histograms. IEEE Trans Systems Man & Cybernetics, 1979; 9(1): 62–66.

[27] Mai C Y, Zheng L H, Xiao C Y, Li M Z. Comparison of apple recognition methods under natural light. Journal of China Agricultural University, 2016; 21(11): 43–50. (in Chinese)

[28] Shi J B, Tomasi C. Good features to track. 1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Seattle: IEEE, 1994; pp.593–600.