

Received January 17, 2020, accepted February 3, 2020, date of publication February 13, 2020, date of current version February 25, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2973757

# Study of Dynamic Tracking Algorithms for Apples Under the Influence of Oscillation

MEI JIANG<sup>1</sup>, QIAN WU<sup>1</sup>, SASHUANG SUN<sup>1</sup>, YUYING SHANG<sup>1</sup>, AND HUAIBO SONG<sup>1</sup>

College of Mechanical and Electronic Engineering, Northwest A&F University, Yangling 712100, China  
Key Lab of Agricultural Internet of Things, Ministry of Agriculture and Rural Affairs, Yangling 712100, China  
Shaanxi Key Laboratory of Agricultural Information Perception and Intelligence Service, Yangling 712100, China

Corresponding author: Huaibo Song (songhuaibo@nwsuaf.edu.cn)

This work was supported by the National Key R&D Program of China (2019YFD1002401) and the National High Technology Research and Development Program of China (863 Program) (No. 2013AA10230402).

**ABSTRACT** Influenced by the force of the wind and agricultural operations, fruits often undergo oscillation, which makes it difficult to automatically monitor their growing status. It is very important to realize dynamic tracking of these oscillating fruits in order to improve automatic monitoring systems and the efficiency of picking robots. In order to investigate the accuracy of the tracking of oscillating fruits, three classic tracking algorithms were adopted and compared: the kernelized correlation filter algorithm (KCF), the compressive tracking algorithm (CT), and the multi-task tracking algorithm (MTT). The effectiveness of these algorithms was verified by testing six video sequences acquired in different environments, and three indices (the average central error, frame loss rate, and time efficiency) were used to verify their performance. The results showed that the KCF algorithm was most appropriate for the tracking of oscillating fruit objects, as it has a lower centering error and a much higher frame rate.

**INDEX TERMS** Object tracking, oscillation, KCF algorithm, CT algorithm, MTT algorithm.

## I. INTRODUCTION

Under natural conditions, fruits are prone to undergoing irregular motion under the influence of the wind and disturbances caused by pruning, grafting and other agricultural operations. This affects the accuracy and efficiency of the automatic monitoring of their growth and robot picking operations. Video tracking technology is used to analyze the target oscillating fruit to obtain the characteristics of its motion, such as the velocity and acceleration of the fruit, in order to generate an accurate estimation of the subsequent motion of the target. It is very important to improve the efficiency of fruit growth monitoring systems and intelligent picking systems.

Fast recognition and accurate positioning are two problems that need to be solved for apple harvesting robots and growth status monitoring systems [1]. For young apples, Gao *et al.* [2] proposed a fruit recognition and location algorithm, which based on the improved connected component labeling algorithm and the shape feature value circularity. Then obtained three-dimensional information by binocular stereo vision system to pick fruits. Zhao *et al.* [3] developed a robotic device for harvesting apples, which consisted of a

manipulator, an end effector and an image-based vision servo control system. A fruit recognition algorithm, which used a support vector machine with a radial basis function, was developed to detect and automatically locate apples within the trees. Liu *et al.* [4] proposed the algorithm based on block classification for apples in plastic bags. The algorithm could reduce the influence of light and the experimental results showed that the false negative rate and the false positive rate were 4.65% and 3.50%.

Many scholars have focused on moving object tracking and have done a lot of work in this area [5], [6]. Lv *et al.* [7] studied a fast method of tracing target fruit for an apple harvesting robot. On the basis of an identification of the first frame image using the Otsu algorithm, the target fruit in the subsequent image frames were traced and recognized using an improved, fast template-matching algorithm involving mean-residual normalized product correlation. The results showed that the recognition time for the designed tracking recognition method was 36% lower than that of the Otsu algorithm. Zhao *et al.* [8] proposed a fast tracking method for overlapping fruits, for which the matching recognition time was 0.185 s without anticipation, and 0.133 s with anticipation. An experimental comparison demonstrated that the proposed method improved the tracking velocity of the

The associate editor coordinating the review of this manuscript and approving it for publication was Gustavo Olague<sup>1</sup>.

robot and made it more practical. Lee [9] proposed a one-shot Siamese network (Siam-OS) to improve the real-time tracking performance of Siam network. The results showed that Siam-OS could achieve fast and effective visual target tracking. Zhang *et al.* [10] proposed a tracking method combining the Kalman filter and data computing process for multiple objects. This method had good tracking performance and the accuracy of this method was as high as 92.9%. Henriques *et al.* [11] used a histogram of oriented gradient features to build a target tracker based on a combination of KCF and a double correlation filter, and achieved good target tracking results. In order to improve the poor performance of compression tracking (CT) algorithm when the objects were occluded, Wang *et al.* [12] proposed an improved CT algorithm based on target segmentation and feature points matching. Chan *et al.* [13] proposed an adaptive CT algorithm that significantly improved conventional CT in four different respects. The results showed that this tracker achieved state-of-the-art performance. For the problem of multi-target continuous tracking, Hosseini *et al.* [14] extended Rao-Blackwellized Monte Carlo Data Association (RBMCDAs) to estimate the number of objects. The modified RBMCDAs had strong adaptability and could be used in many different situations.

With the improvement in agricultural informatization, the demand for real-time monitoring of fruit growth information and automatic picking is becoming increasingly urgent. Fruits usually grow in complicated, unstructured environments, and they are often in an oscillating state and affected by the surrounding branches and leaves, making automatic tracking and monitoring of them difficult. Windy and rainy weather will cause fruit oscillation movement and bring difficulties to fruit monitoring. The accuracy of tracking for individual oscillatory fruits is indeterminate. Due to illumination changes and occlusion caused by irregular movement and oscillation of fruits, the Kalman filter and optical flow method are not suitable for fruit tracking. Based on video analysis technology, the objectives of this research are to test the performance of three popular object tracking algorithms (the KCF algorithm, the CT algorithm, and the MTT algorithm) to obtain velocity and acceleration curves for the oscillating fruits in order to enable fast and accurate tracking and to lay the foundation for the establishment of intelligent monitoring systems for fruit growth. The test results also have significance in terms of improving the efficiency of picking robots.

## II. MATERIALS AND METHODS

### A. MATERIALS

The videos of oscillating fruits used in this study were taken on September 23, 2016 and May 23, 2017, when the weather was fine and the wind grade was 2. The sites at which the videos were shot were at the Horticultural College of Northwest A&F University and the Economic Tree Garden in the south campus of Northwest A&F University. The videos were taken by a cell phone (OPPO R7, 1.7GHz, RAM 3GB,

13 million pixels, Guang Dong OPPO Mobile Telecommunications corp., Ltd, Guangdong, China) and the Nikon D90 camera (12.3 million effective pixels, Nikon Corporation, Tokyo Metropolitan, Japan).

Frame loss in a camera can happen due to frames getting dropped at the sensor, camera driver or application. These videos were shot using the mobile phone and Nikon D90 camera with frame rates of 29 and 24 frames per second, respectively. The possibility of camera frame loss is extremely low. Fruits didn't oscillate at a high speed in the videos, and the difference between the  $t^{th}$  frame and the  $(t+1)^{th}$  frame or the  $(t-1)^{th}$  frame was small. If the  $t^{th}$  frame was lost, it could be replaced by the  $(t+1)^{th}$  frame or the  $(t-1)^{th}$  frame.

In order to show the state of fruit oscillation, the fruits in the initial frame should be in the center of the lens. For each test video, we moved and mounted the camera to locate the fruit in the center of the lens field of view according to the wind direction and wind pressure.

Six experimental video sequences were obtained for different oscillation frequencies, and the resolution of the video images was  $360 \times 240$  pixels. The specific information obtained from the video is shown in Table 1. The test videos contained different influencing factors such as motion blur, out-of-view, illumination variation, occlusion and scale variation. These factors seriously affected the performance of target tracking. These six videos were representative and fully showed the performance of the three algorithms from many aspects.

TABLE 1. Test video information.

Video sequence	Total frames	Main influencing factors
Video 1	100	No
Video 2	100	Illumination variation
Video 3	100	Target is partially occluded
Video 4	100	Motion blur, Out-of-view, Illumination variation, and Scale variation
Video 5	100	Illumination variation, Scale variation
Video 6	100	Illumination variation

All the procedures were run in the MATLAB R2016a environment, and the hardware used was a Lenovo Z40 laptop with 4GB RAM and 2.4 GHz dominant frequency.

### B. KCF TRACKING ALGORITHM

The KCF algorithm is based on a circulant structure that uses tracking by detection with kernels, and relies on a histogram of oriented gradient (HOG) feature instead of the original grayscale feature. The correlation filter is extended from a single channel to a multi-channel scheme [11], [15]. The core of the KCF tracking algorithm involves cyclic shifting of the training samples. The tracked target is a positive sample, and the rest of the surrounding environment is a negative sample, and a training discriminant classifier is constructed based on

this scheme [11]. The similarity between the target region and the candidate region is calculated using a kernel function. The candidate region with the largest similarity is selected as the new tracking target. The algorithm is accelerated using a Fourier transform, which greatly improves the tracking efficiency of the algorithm. The specific process is as follows:

(1) A kernel ridge regression classifier is used as the core, and the target region is shifted using the circulant matrix theory. A large number of training samples are constructed for the classifier.

(2) Training the classifier: The classifier is used to calculate the probability that all candidate regions become the target area, and the candidate region with the largest probability is selected as the tracking target.

(3) Fast detection and appearance model updates. The new input image is cyclically shifted to construct a candidate sample set, and the region corresponding to the element with the largest probability is identified as the tracking target. After this fast detection, the new target area is cyclically shifted to construct a new training sample set, and the model parameters used in the next frame for the classifier detection process are updated.

### C. CT TRACKING ALGORITHM

Compressed sensing is a new research field that is based on signal sparse representation and approximation theory [16]–[18]. It makes full use of the sparse characteristics of the target signal structure through low-resolution, non-correlated measurements of Nyquist sampling data to perceive high-dimensional sparse signals [19]. The core of the CT tracking algorithm involves creating a low-dimensional compressed subspace by projecting the original image feature space with a very sparse measurement matrix under a full restricted isometry property (RIP) condition [20]. The low-dimensional compressed subspace can effectively preserve the information from the high-dimensional image feature space. The sparse measurement matrix extracts the foreground and background characteristics as positive and negative samples to update the classifier for online learning, and then uses a naive Bayes classifier to evaluate candidate samples, where the candidate sample with highest probability is the target for tracking. More specifically, the process is as follows:

(1) In the  $t^{\text{th}}$  frame, a number of images (positive samples) and backgrounds (negative samples) are obtained by sampling, and multi-scale transformations are performed on them. The dimensionality of the multi-scale image is reduced using a sparse measurement matrix, and the dimension reduction feature is used for training after the simple Bayesian classifier is applied.

(2) In the  $(t+1)^{\text{th}}$  frame, the  $N$  scanning window is tracked at the target position obtained in the previous frame. Using the sparse measurement matrix to reduce its dimensions, the features are extracted, and the  $t^{\text{th}}$  frame is used to train the naive Bayes classifier to classify the target window. The window with the highest classification score is considered to be the

target window. This achieves target tracking from the  $t^{\text{th}}$  frame to the  $(t+1)^{\text{th}}$  frame.

### D. MTT TRACKING ALGORITHM

The MTT tracking algorithm is based on the particle filter framework. Target tracking is considered as a multi-task sparse learning problem, and the particle model is used for the dynamic update of the dictionary template linear combination [21], [22]. The representation of each particle in the MTT is learned as a single task, and the learning problem can be solved using a method of accelerating the gradient (APG), which effectively resolves the closure of the yield sequence. Thus, MTT is computationally attractive [23], [24]. The specific process is as follows:

(1) Tracking the multi-task representation of the target: In the multi-task learning (MTL) framework, tasks that share dependencies in terms of features or learning parameters are jointly solved in order to capitalize on their inherent relationships. The tracking problem is defined as an MTL problem in which the representation of the learning particle is treated as a single task. In general, the particles in the trace are calculated independently. In the current tracking state, the particles are randomly sampled around the current state of the tracking object using a zero mean Gaussian distribution.

(2) Imposing conditions of joint sparsity and mixed norm: Joint sparsity encourages all particle representations to be individually sparse and to share the same dictionary templates. An  $l_{p,q}$  mixed norm is used for the reconstruction error.

### E. OBTAIN THE TRAJECTORY OF OSCILLATING FRUIT

Based on the accurate tracking of oscillating fruits, kinematic parameters such as the velocity and acceleration of the target can be obtained by analyzing and calculating the trajectory data. This is important in terms of obtaining the best observation data frame to realize the intelligent monitoring of fruit growth information and predict the trajectory required from the picking robot.

#### 1) TARGET VELOCITY

We assume that the positions of the moving target at times  $t_1$  and  $t_2$  ( $t_2 > t_1$ ) are  $p_1 = (x_1, y_1)$  and  $p_2 = (x_2, y_2)$ , respectively. The velocity of the moving target at time  $t_2$  can be calculated by Eq. (1):

$$V = \frac{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}}{t_2 - t_1} \quad (1)$$

where  $V$  represents the velocity of the moving target.

#### 2) TARGET ACCELERATION

We assume the moving target has velocity  $V_1$  and  $V_2$  at times  $t_1$  and  $t_2$ , respectively. Then the acceleration  $a$  of the moving target at time  $t_2$  can be calculated by Eq. (2):

$$a = \frac{V_2 - V_1}{t_2 - t_1} \quad (2)$$

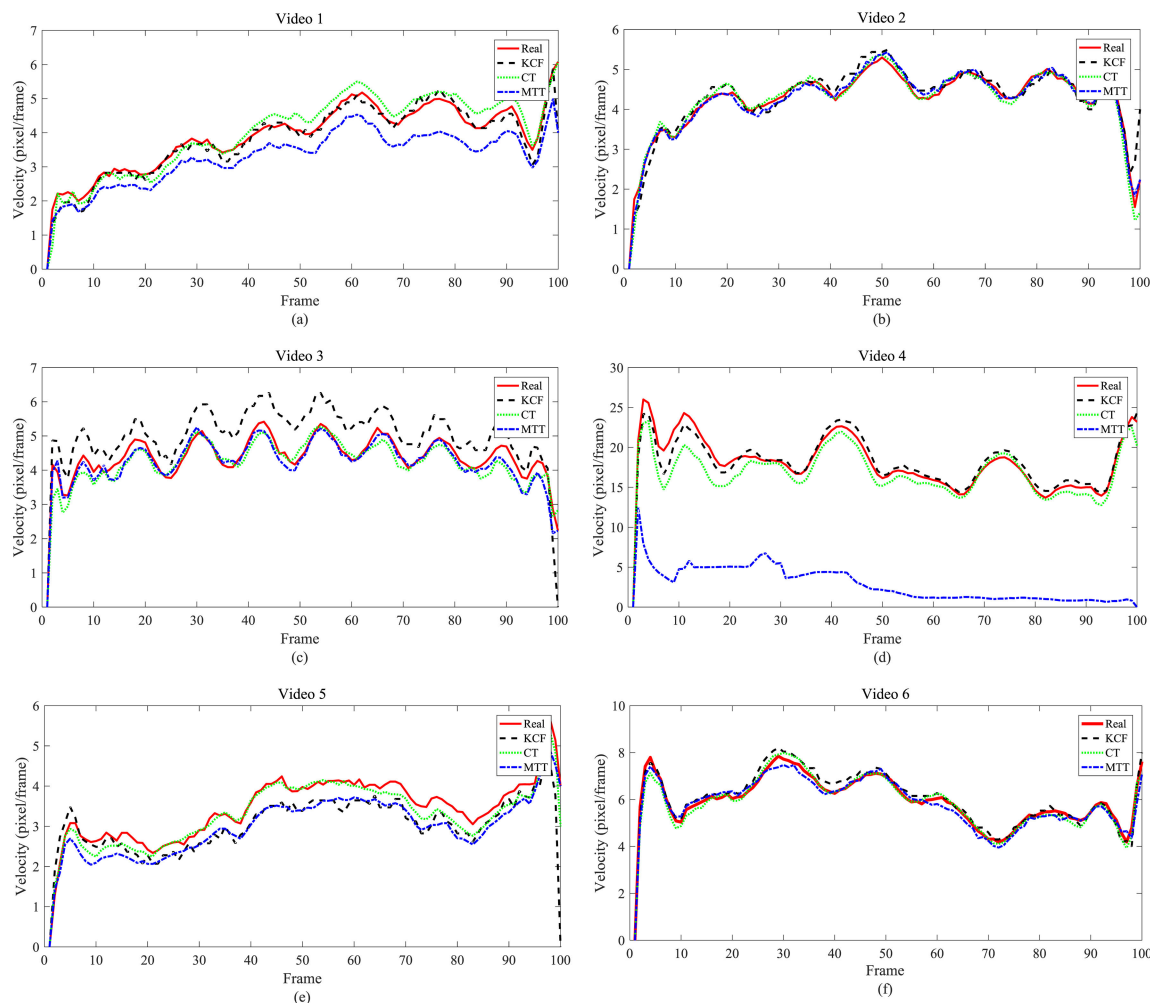


FIGURE 1. Velocity curve of oscillating fruit: (a) Video 1; (b) Video 2; (c) Video 3; (d) Video 4; (e) Video 5; (f) Video 6.

This acceleration can take positive and negative values, representing an increase or decrease in the velocity, respectively. The target’s acceleration represents the rate of change of its velocity.

**F. EVALUATION INDICATORS**

In order to further objectively evaluate the three algorithms selected in this research, we used the center point position error, the frame loss rate, and the number of frames per second as criteria to quantitatively evaluate the algorithms for the six videos.

(1) The center point position error is the square of the error between the central position of the tracked target and the true target center position, as shown in Eq. (3):

$$Error = \sqrt{(x_g - x_t)^2 + (y_g - y_t)^2} \tag{3}$$

where  $x_g$  and  $y_g$  represent the exact position of the manually calibrated target fruit, and  $x_t$  and  $y_t$  represent the center coordinates of the tracked target. The smaller the error, the higher the accuracy of the target tracking.

(2) The frame loss rate is the ratio of the number of frames for which the coincidence of the resulting tracked area is less than 50% with the actual target area and the total frames. The smaller the frame loss rate, the better the algorithm.

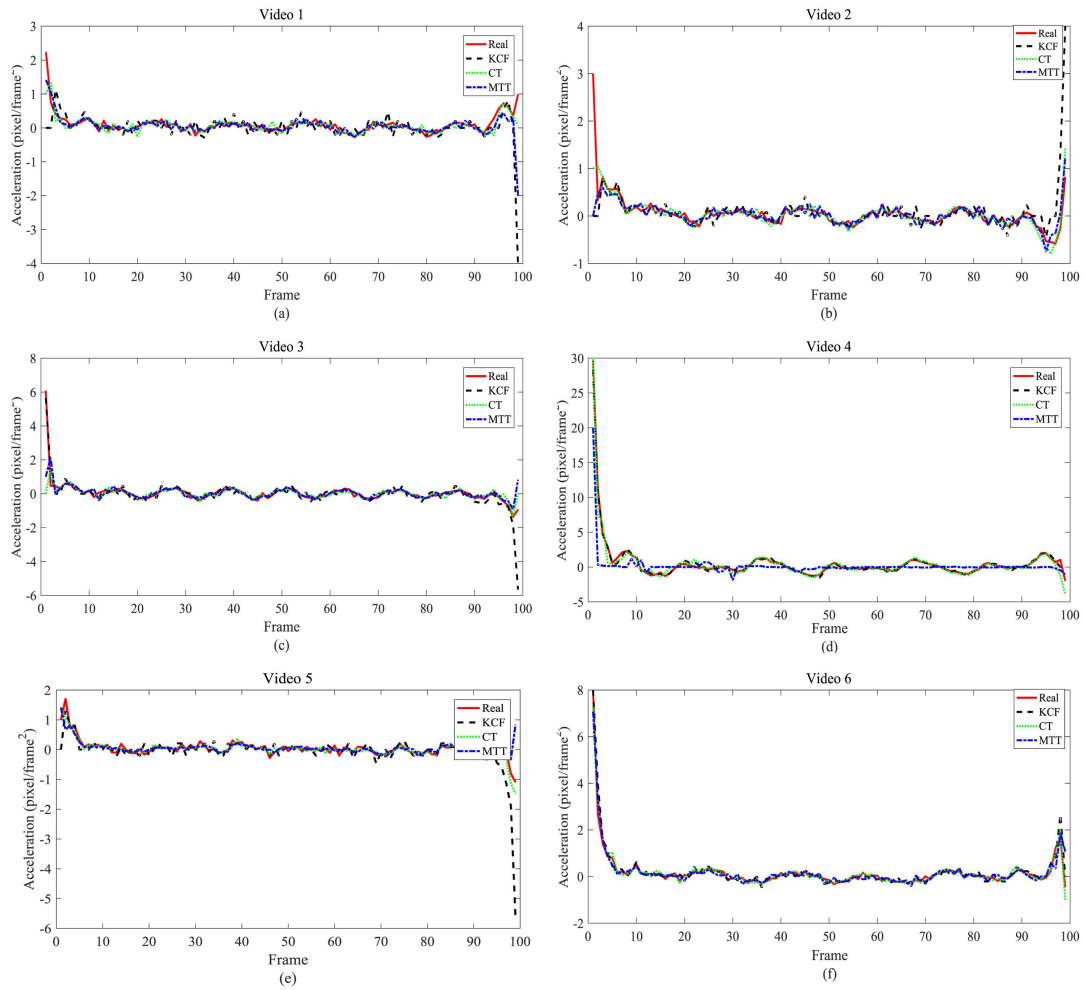
**III. RESULTS AND DISCUSSION**

In this research, six experimental videos with different oscillation frequencies were used. The six groups of videos included challenging problems such as local occlusion, variations in illumination and scale, and the target moving outside the field of view. In this manuscript, the KCF, CT, and MTT algorithms were used to track the oscillating fruit.

**A. QUANTITATIVE RESULTS AND DISCUSSION**

The velocity curve for the oscillation of the fruit is shown in Figure. 1, where the abscissa is the number of frames, and the ordinate is the velocity/pixel.

The red line represents the true velocity of the motion of the oscillating fruit; the purple line indicates the velocity curve obtained using the KCF algorithm; the green line indicates the



**FIGURE 2.** Acceleration curve of oscillating fruit: (a) Video 1; (b) Video 2; (c) Video 3; (d) Video 4; (e) Video 5; (f) Video 6.

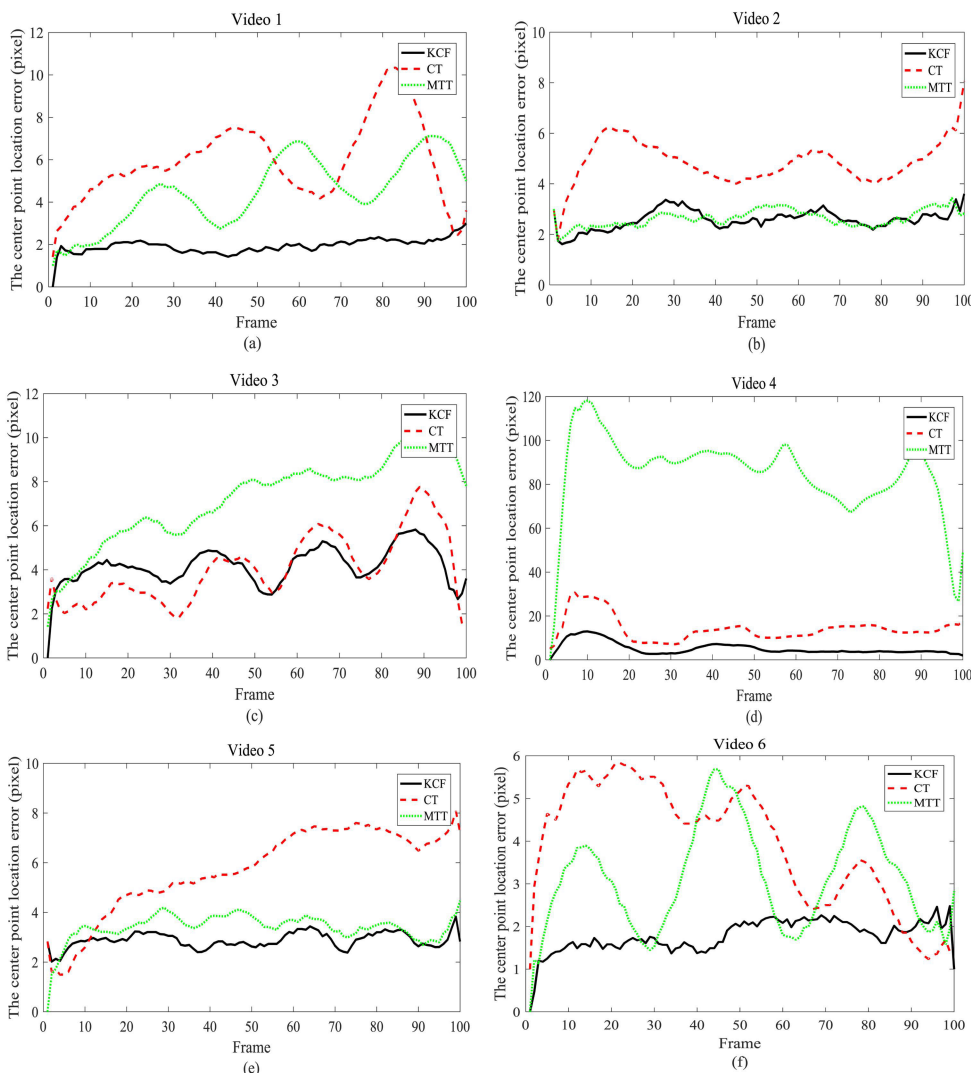
velocity curve obtained using the CT algorithm; and the blue line indicates the velocity curve using the MTT algorithm. In the process of fruit picking, fruits with a higher speed of movement are relatively difficult to pick, and it is much easier to pick them when they are at rest. From the velocity curve in Figure. 1, the robot can choose a velocity of zero to locate the oscillating fruits in the picking operation, in order to improve the efficiency of picking. There was almost no motion blur in the lower velocity range; this was beneficial in giving a clear monitoring image, in order to allow for accurate target monitoring.

Table 2 shows the correlation coefficients between the tracked velocity and the real velocity of the oscillating fruit obtained by the three algorithms, where the closer the absolute value of the correlation coefficient to one, the more similar the two curves. It can be seen from Table 2 that the average value of the correlation coefficient for the KCF algorithm was 0.94, for the CT algorithm the value was 0.97, and for the MTT algorithm this was 0.91, thus indicating that the acceleration obtained using the CT algorithm was closest to the real velocity.

**TABLE 2.** Velocity correlation coefficient.

Video sequence	KCF algorithm	CT algorithm	MTT algorithm
Video 1	0.97	0.98	0.98
Video 2	0.95	0.99	0.99
Video 3	0.93	0.94	0.96
Video 4	0.97	0.93	0.55
Video 5	0.82	0.98	0.98
Video 6	0.99	0.99	0.98
Average	0.94	0.97	0.91

The acceleration curve for the oscillating fruit is shown in Figure. 2, where the abscissa is the number of frames, and the ordinate is the acceleration/pixel. The red line indicates the true velocity of motion of the oscillating fruit; the purple line indicates the acceleration curve obtained using the KCF algorithm; the green line indicates value from the CT algorithm; and the blue line indicates the value from the MTT algorithm.



**FIGURE 3.** Error curves of tracking the oscillating apples. (a) Video 1; (b) Video 2; (c) Video 3; (d) Video 4; (e) Video 5; (f) Video 6.

Table 3 shows the correlation coefficient between the tracked acceleration and the real acceleration of the oscillating fruit obtained by the three algorithms, where the closer the absolute value of the correlation coefficient is to one, the more similar the two curves. It can be seen from Table 3 that the average value of the correlation coefficient obtained using the KCF algorithm was 0.65; the average value obtained using the CT algorithm was 0.78; and the average value for the MTT algorithm was 0.64. Hence, the acceleration obtained from the CT algorithm was closest to the real acceleration.

The center point position error of the three algorithms is shown in Figure. 3, where the abscissa is the number of frames, and the ordinate is the center point position error. The black, red and green lines represent the curve of the mean square error between the coordinates of the center point of each frame and the coordinates of the true center point, for the KCF algorithm, CT algorithm and MTT algorithm, respectively. The average center errors for each

**TABLE 3.** Acceleration correlation coefficient.

Video sequence	KCF algorithm	CT algorithm	MTT algorithm
Video 1	-0.13	0.74	0.40
Video 2	0.32	0.77	0.50
Video 3	0.81	0.32	0.51
Video 4	0.99	0.98	0.90
Video 5	0.66	0.90	0.54
Video 6	0.97	0.99	0.96
Average	0.65	0.78	0.64

video sequence are shown in Table 4. It can be seen that the KCF algorithm achieved the best tracking accuracy for the six video sequences, with an average center point error of 3.13 pixels. The CT algorithm achieved good tracking for five videos, and the tracking was best in Video 4, with an average center point error of 6.53 pixels. The MTT algorithm



**FIGURE 4.** Comparison of three algorithmic tracking results. (a) Video 1; (b) Video 2; (c) Video 3; (d) Video 4; (e) Video 5; (f) Video 6.

achieved good tracking for five videos, although the tracking effect for Video 4 was relatively poor, with an average center error of 17.98 pixels.

The frame loss rates for each algorithm for the different videos are shown in Table 5. It can be seen from Table 5 that the frame loss rates were 0, 1.67% and 16.33% for the KCF, CT and MTT algorithms, respectively. It can therefore be

concluded that the tracking effect in the KCF algorithm was the most stable, the CT algorithm was second, and the MTT algorithm was the least stable.

The average frame rates of the three algorithms running on the different video sequences are shown in Table 6. The data were obtained by running the algorithm 10 times for each video sequence, and the average values were used. The results

**TABLE 4. Average center error.**

Video sequence	KCF(pixel)	CT(pixel)	MTT(pixel)
Video 1	1.94	5.99	4.51
Video 2	2.57	4.87	2.65
Video 3	4.22	4.11	7.30
Video 4	5.36	14.56	86.89
Video 5	2.87	5.69	3.42
Video 6	1.80	3.95	3.14
Average	3.13	6.53	17.98

**TABLE 5. Tracks the frame loss rate.**

Video sequence	Total number of frames	KCF(%)	CT(%)	MTT(%)
Video 1	100	0	0	0
Video 2	100	0	0	0
Video 3	100	0	0	0
Video 4	100	0	0	0
Video 5	100	0	10	98
Video 6	100	0	0	0
Average	100	0	1.67	16.33

show that the average frame rate for target tracking in the KCF algorithm was 24.73 f/s, and since the running speed was higher than that of the other algorithms, the algorithm can give real-time results. The average frame rate for target tracking by the CT algorithm was 12.67 f/s, and its average speed was 51.23% of KCF algorithm's average speed, which means that it can basically provide real-time results. The average frame rate in the MTT algorithm was 0.36 f/s, which cannot generate real-time results, and this approach was therefore unsuitable for dynamic tracking of the oscillating fruit.

**TABLE 6. Average frame rate.**

Algorithm	Maximum(f/s)	Minimum(f/s)	Average(f/s)
KCF	30.05	19.69	24.73
CT	14.21	10.10	12.67
MTT	0.42	0.27	0.36

## B. QUALITATIVE RESULTS AND ANALYSIS

The tracking results for the oscillating fruit are shown in Figure. 4. The red tracking box shows the results for the KCF algorithm, the green box for the CT algorithm, and the blue box for the MTT algorithm. Four frames were randomly selected in each video sequence. It can be seen from Figure. 4(a)-(f) that there were variations in illumination and slight variations in scale. The results showed that all three algorithms could accurately track the oscillated fruits, indicating that changes in illumination and scale had little effect on tracking. In Figure. 4c, the oscillating fruit was occluded by leaves, but the tracking by the three algorithms did not encounter serious problems, and it was still possible to

accurately track the oscillating fruits. In Figure. 4d, the oscillating fruit becomes blurred as a result of faster movement, and there were variations in scale and the problem of the target outside the field of view at the same time. It can be seen that the KCF algorithm was not affected by these factors, and could accurately track the target. Use of the CT algorithm had little effect on tracking, except in the case of the losing the target from the field of vision, although when the target moved back into the field of vision, tracking was resumed. The MTT algorithm was strongly affected by these factors, and the tracking target was completely lost from the third frame onwards.

In summary, for the purposes of tracking the oscillating fruit, the KCF algorithm was the best for a variety of scenes, and could track the target accurately without requiring tuning of the parameters. The CT algorithm tracking was also good, with a high frame rate, but suffered from over-reliance on the choice of parameters. Although in general the MTT algorithm could accurately track the target, the tracking effect was poor when it encountered more challenging problems. It was also slow, and was therefore unsuitable for tracking oscillating fruits.

## IV. CONCLUSION

The tracking of oscillating fruit has great research significance in terms of obtaining the optimal target monitoring frame and determining the required trajectory for a picking robot. In this study, we applied the commonly used KCF, CT, and MTT algorithms to the accurate and efficient tracking of oscillating apples, and the main conclusions were as follows:

(1) The KCF algorithm had high accuracy when tracking the moving target in the case where the video sequence contained challenging tracking problems. The average center error was 3.13 pixels, the frame loss rate was zero. The processing speed was fast, with an average frame rate of 24.73 f/s, and this ensures real-time performance of the algorithm, meaning that it can be applied to the tracking of oscillating fruit targets.

(2) The CT algorithm gave velocity and acceleration curves that were closer to the real value, but relied too much on the selected parameters, meaning that the target could be tracked accurately only when the parameters were selected appropriately.

(3) If there were no tracking problems such as motion blur, scale variation, or the target moving out of the field of view, the MTT algorithm achieved suitable tracking. However, when there were challenging problems such as motion blur, out of view, and scale variation, it lost the tracking target, leading to a failure of tracking. Moreover, the algorithm runs at a low speed, with an average frame rate of 0.36 f/s, thus falling below real-time performance.

(4) Three algorithms were used to track fruits and analyze motion trajectories. However, serious occlusion and high-speed fruit oscillation may lead to errors in the fruit motion tracking analysis.



Deep learning plays an important role in the field of image processing. It can reduce the influence of various environmental factors such as illumination, occlusion, oscillation and other factors and make the algorithm more robust. In the future, more attentions should be paid to deep learning and use deep learning to solve object tracking problems.

## ACKNOWLEDGMENT

The authors would like to thank all of the authors cited in this article and the anonymous referees for their helpful comments and suggestions.

## REFERENCES

- [1] D. Wang, H. Song, and D. He, "Research advance on vision system of apple picking robot," *Trans. Chin. Soc. Agricult. Eng.*, vol. 33, no. 10, pp. 59–69, May 2017.
- [2] H. Gao, Y. Liu, D. Li, and Y. Yu, "Vision localization algorithms for apple bagging robot," in *Proc. 29th Chin. Control Decis. Conf. (CCDC)*, May 2017, pp. 1948–9447.
- [3] Z. De-An, L. Jidong, J. Wei, Z. Ying, and C. Yu, "Design and control of an apple harvesting robot," *Biosyst. Eng.*, vol. 110, no. 2, pp. 112–122, Oct. 2011.
- [4] X. Liu, W. Jia, C. Ruan, D. Zhao, Y. Gu, and W. Chen, "The recognition of apple fruits in plastic bags based on block classification," *Precis. Agricult.*, vol. 19, no. 4, pp. 735–749, Aug. 2018.
- [5] T. Xu, Z.-H. Feng, X.-J. Wu, and J. Kittler, "Learning adaptive discriminative correlation filters via temporal consistency preserving spatial feature selection for robust visual object tracking," *IEEE Trans. Image Process.*, vol. 28, no. 11, pp. 5596–5609, Nov. 2019.
- [6] Y. Ge, Y. Xiong, G. L. Tenorio, and P. J. From, "Fruit localization and environment perception for strawberry harvesting robots," *IEEE Access*, vol. 7, pp. 147642–147652, 2019.
- [7] J. Lv, D. Zhao, and W. Ji, "Fast tracing recognition method of target fruit for apple harvesting robot," *Trans. Chin. Soc. Agricult. Machinery*, vol. 45, no. 1, pp. 65–72, Apr. 2014.
- [8] D. Zhao, T. Shen, Y. Chen, and W. Jia, "Fast tracking and recognition of overlapping fruit for apple harvesting robot," *Trans. Chin. Soc. Agricult. Eng.*, vol. 31, no. 2, pp. 22–28, Jan. 2015.
- [9] D.-H. Lee, "One-shot scale and angle estimation for fast visual object tracking," *IEEE Access*, vol. 7, pp. 55477–55484, 2019.
- [10] W. Zhang, S. Cho, J. Chae, Y. Sung, and K. Cho, "Object tracking method based on data computing," *J. Supercomput.*, vol. 75, no. 6, pp. 3217–3228, Jun. 2019.
- [11] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, Mar. 2015.
- [12] W. Wang, M. Jiang, Y. Yan, X. Chen, and W. Zhao, "Improved CT algorithm based on target block division and feature points matching," *EURASIP J. Image Video Process.*, vol. 2018, no. 1, pp. 1–18, Dec. 2018.
- [13] S. Chan, X. Zhou, J. Li, and S. Chen, "Adaptive compressive tracking based on locality sensitive histograms," *Pattern Recognit.*, vol. 72, pp. 517–531, Dec. 2017.
- [14] S. S. Hosseini, M. M. Jamali, and S. Särkkä, "Variational Bayesian adaptation of noise covariances in multiple target tracking problems," *Measurement*, vol. 122, pp. 14–19, Jul. 2018.
- [15] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2005, pp. 886–893.
- [16] R. Baraniuk, "Compressive sensing [lecture notes]," *IEEE Signal Process. Mag.*, vol. 24, no. 4, pp. 118–121, Jul. 2007.
- [17] E. J. Candès, J. K. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Commun. Pure Appl. Math.*, vol. 59, no. 8, pp. 1207–1223, Aug. 2006.
- [18] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.
- [19] H. M. Yu and G. Y. Fang, "Research on compressive sensing based 3D imaging method applied to ground penetrating radar," *J. Electr. Inf. Technol.*, vol. 32, no. 1, pp. 12–16, Jan. 2010.
- [20] E. J. Candès and T. Tao, "Decoding by linear programming," *IEEE Trans. Inf. Theory*, vol. 51, no. 12, pp. 4203–4215, Dec. 2005.

- [21] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, "Robust visual tracking via multi-task sparse learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2042–2049.
- [22] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, "Robust visual tracking via structured multi-task sparse learning," *Int. J. Comput. Vis.*, vol. 101, no. 2, pp. 367–383, Jan. 2013.
- [23] A. Quattoni, X. Carreras, M. Collins, and T. Darrell, "An efficient projection for L1, infinity regularization," in *Proc. ICML*, Jan. 2009, pp. 857–864.
- [24] X.-T. Yuan, X. Liu, and S. Yan, "Visual classification with multitask joint sparse representation," *IEEE Trans. Image Process.*, vol. 21, no. 10, pp. 4349–4360, Oct. 2012.



**MEI JIANG** was born in Xi'an, China, in 1996. She received the B.S. degree from Northwest A&F University (NWAUFU), in 2019, where she is currently pursuing the M.S. degree in agricultural electrification and automation. Her research interests include image processing, deep learning, and saliency detection.



**QIAN WU** received the B.S. and M.S. degrees in agricultural electrification and automation from Northwest A&F University (NWAUFU), Yangling, China, in 2016 and 2019, respectively. Her research interests include object detection, segmentation, semantic segmentation, and image processing.



**SASHUANG SUN** received the M.S. degree from Northwest A&F University, Yangling, China, in 2019. She is currently pursuing the Ph.D. degree with the College of Biosystems Engineering and Food Science, Zhejiang University. Her research interest includes agricultural image processing.



**YUYING SHANG** was born in Zhengzhou, China, in 1999. She is currently pursuing the B.S. degree in electronic information engineering from Northwest A&F University. Her research interests include computer vision and image processing.



**HUAIBO SONG** received the M.S. and Ph.D. degrees in mechanical and electrical engineering from Shandong University, Jinan, China, in 2009. He was a Visiting Research Student with Texas A&M University, from 2014 to 2015. He is currently an Associate Professor with the College of Mechanical and Electrical Engineering, Northwest A&F University, Yangling, China. His current research interests include object detection, segmentation, and semantic segmentation.