Hindawi
Journal of Sensors
Volume 2021, Article ID 3353032, 10 pages
https://doi.org/10.1155/2021/3353032

Research Article

An Improved Camshift Tracking Algorithm Based on LiDAR Sensor

Yong Lv[1] **and Hairong Zhu**[2]

[1]*School of Mechanical and Electric Engineering, Soochow University, Suzhou, China*
[2]*School of Mechanical and Electrical Engineering, Jiangsu College of Engineering and Technology, Nantong, China*

Correspondence should be addressed to Hairong Zhu; hairongzhu@163.com

Received 27 May 2021; Revised 19 October 2021; Accepted 25 October 2021; Published 12 November 2021

Academic Editor: Aijun Yin

Aiming at the problems of inaccurate interaction point position, interaction point drift, and interaction feedback delay in the process of LiDAR sensor signal processing interactive system, a target tracking algorithm is proposed by combining LiDAR depth image information with color images. The algorithm first fuses the gesture detection results of the LiDAR and the visual image and uses the color information fusion algorithm of the Camshift algorithm to realize the tracking of the moving target. The experimental results show that the multi-information fusion tracking algorithm based on this paper has achieved higher recognition rate and better stability and robustness than the traditional fusion tracking algorithm.

1. Introduction

Target tracking [1] is one of the most important research directions in the field of computer vision, and its application areas include surveillance, medical imaging, and human-computer interaction. Although target tracking has been studied for many years and important progress has been made, there are still many problems. Generally, the interaction system based on LiDAR sensor is affected by the following factors: such as occlusion, complex environment, and illumination variation. In addition, it is crucial to improve the real-time performance of LiDAR sensor for data processing.

From the perspective of the sensor, target tracking algorithms can be roughly divided into the following categories: vision-based target tracking [2–4], LiDAR-based target tracking [5–7], and target tracking based on the fusion of vision and LiDAR [8–10]. For vision-based target tracking methods, it often cannot provide distance information of moving objects, and the effect is not good in object recognition and tracking [11, 12]. With the popularization of stereo vision, we can obtain distance information of moving objects, but the amount of calculation is relatively large [13–15]. In recent years, due to the advantages of high-ranging accuracy and real-time performance, LiDAR sensor has attracted more and more attention in the target tracking of interactive systems [16, 17]. LiDAR-based target tracking usually uses traditional tracking methods, which can continuously and accurately obtain measurement data and predict the state of the target [8]. Furthermore, in order to improve the accuracy of tracking, the LiDAR-based target tracking combines statistics, random decision theory, and intelligent control calculation methods to judge the tracking target according to certain rules [18].

Since the methods of vision-based and LiDAR-based target tracking have their own advantages and disadvantages, the fusion of the above two methods can further improve the tracking effect. Therefore, Broggi et al. [19] used LiDAR and stereo vision to draw raster maps and merged the information of the two. However, the above method is limited to a specific LiDAR sensor, and feature-based methods are used in the tracking process to reduce the amount of calculation. Subsequently, Petrovskaya and Thrun [20] proposed a method to map the depth information of LiDAR to an image sequence based on the polar coordinate system and detect moving targets through the difference between two consecutive frames. However, this method only considers the depth data information of the last few frames, and it ignores the state prediction fusion and may not be able to detect and track the target when facing some complex scenes. Other typical methods include using Kalman filter for target tracking [21].

In this paper, the tracking algorithm is based on the LiDAR sensor and the visual tracking is integrated. As we all know, once the target is blocked or interfered by environmental changes, the LiDAR sensor will not be able to continuously measure the distance of the object, which causes problems such as inaccurate interaction point position, interaction point drift, and interaction feedback delay. In response to the above problems, Guo et al. [22] developed a high-throughput crop phenotyping platform that integrates LiDAR sensors, high-resolution cameras, thermal imaging cameras, and hyperspectral imagers. It combines LiDAR and traditional remote sensing technologies to obtain higher-precision three-dimensional data. Rossi et al. [23] studied the application of differential absorption LiDAR in monitoring the abnormal concentration of chemical substances in the atmosphere, by using a multiwavelength method to improve the accuracy of atmospheric gas concentration measurement. Chen et al. [24] combined time-correlated single photon counting (TCSPC) technology and LiDAR and proposed a three-beam TCSPC LiDAR system, which effectively improves the ranging accuracy of the TCSPC LiDAR system.

In addition, the real-time performance of data processing in LiDAR sensor is also an important issue that must be resolved. For example, Luo et al. [25] proposed a real-time ground segmentation method based on probability occupancy grids. When there is occlusion, the LiDAR sensor is used to solve the environmental perception task, thereby reducing the processing scale of data and reducing the calculation time. Lyu et al. [26] proposed a field programmable gate array design based on a segmented algorithm of convolutional neural network and applied this method to the segmentation of the drivable area of autonomous driving. It has proved that the algorithm can process LiDAR data in real-time.

In order to take into account both the accuracy and efficiency of LiDAR sensor data processing, this paper proposes a target tracking algorithm, and the main contributions of this paper are listed as follows:

(1) A fusion strategy that combines the depth image information of LiDAR with color images, and put forward a fitting factor that varies with the degree of occlusion

(2) An improved Camshift tracking algorithm based on Kalman filter, which improves the robustness of tracking in a complex environment through the method of linear fit

2. Gesture Detection Information Fusion Framework

In this paper, LiDAR sensor operation recognition is used for data interaction. At the same time, we use a computer to accurately capture the users' command actions and provide real-time and accurate feedback to achieve human-computer interaction. The system interaction frame structure diagram is shown in Figure 1. The user stands in front of the interactive screen and the LiDAR sensor, slides (without contact) on the front of the screen, and clicks on the relevant interactive content, which can replace and implement most of the functions of the mouse, including clicking, sliding, and dragging. The proposed system supports multiple users for real-time gesture interaction.

Before performing target tracking, this paper first proposes a gesture detection fusion method, which combines the gesture detection results of LiDAR and visual images. On the one hand, the LiDAR is continuously measured and calculated to realize the distinction of moving regions; on the other hand, the camera uses the continuous frame difference method to realize the detection of moving targets; then, the detection results of the two are integrated. Finally, the gesture detection result is given by the two sensors and the detection results of multiple frames of images. The specific implementation method will be described in detail below.

Based on the moving target $t = \{obj_i = (f, b_L, b_c), i = 1, 2, \cdots, N\}$, for each image frame $b_c i$, a score $s_c i$ is given with reference to the gesture detector. The multiframe detection result of the gesture detector is $S_c = \{s_c i, i = 1, 2, \cdots, N\}$. If the tracking target is more stable, the probability that it is a target independent of the background is greater. The cost Γ associated with the preceding and following frames is used as the basis S_L for judging whether it is a gesture, $S_L = \{s_l^i = 1 - \Gamma i, i = 1, 2, \cdots, N\}$. When $i = 1$, it is considered that there is no associated cost during initialization, so $s_l^i = 1$. When the association fails, $s_l^i = -1$.

For the detection results of the fusion image, we propose a voting fusion mechanism. The gesture tracking score result is obtained based on formula (1). When $S_F > \Gamma 1$, it is determined as a gesture.

$$S_F = W[\sigma(S_c), S_L]^T. \tag{1}$$

In the above formula, W is the weight, and σ is a function that projects S_c into $[-1, 1]$, which can be expressed as the following formula.

$$\sigma(s) = \frac{2}{1 + \exp(-s/4)} - 1. \tag{2}$$

Finally, the exit rules are defined. We define the exit conditions for the score S_F and the number of survival frames of the tracking target N_{birth}, as shown in formula (3). If the tracking image frame number is greater than the maximum defined number of frames, the tracking target still cannot be determined as a gesture, then, exit the tracking list.

$$\begin{cases} N_{\text{birth}} > \tau_n, \\ S_F < \tau_2, \end{cases} \tag{3}$$

where τ is the defined maximum number of frames.

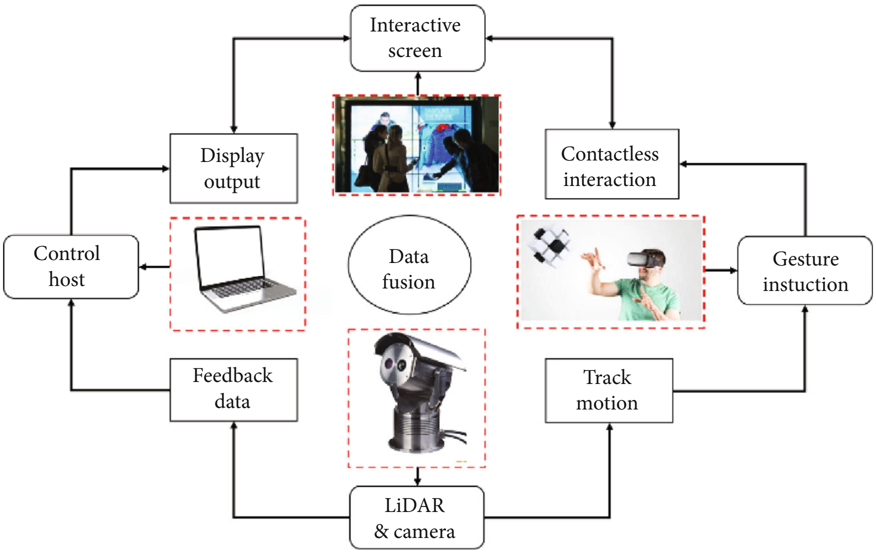


Figure 1: Schematic diagram of system structure.

3. LiDAR Depth Information and Visual Image Information Fusion Algorithm

LiDAR is used to construct position coordinates by scanning data from the target surface. In an interactive system, if the tracking target is blocked or disappears from the effective laser plane instantly, the scan often fails. When the above situation occurs, the information obtained by the LiDAR is not comprehensive, which will cause the problems of the interaction point drift and the unsmooth interaction. Therefore, in this section, we propose a method that combines the color information of the Camshift algorithm to track moving targets. Figure 2 is a flow chart of Camshift algorithm information fusion control.

As shown in Figure 2, first, we fused the data obtained by the LiDAR and the camera. Second, we transformed the space coordinates of the LiDAR to pixel coordinates and ensure time synchronization. Finally, we completed the gesture detection through feature extraction and fusion. Further, we performed noise reduction processing and used the back projection method to obtain the color histogram of the target area. After completing the above steps, the Meanshift algorithm was used for calculation, and then continuous tracking was achieved through window iteration.

In order to facilitate understanding, this article further explained the object tracking process. When the image information was denoised, the window initialization processed was performed, and the color space of the image in the area was converted from RGB to HSV, then the H value of each pixel was sampled and counted. On this basis, the method of histogram statistics was used to replace the pixel value with the color probability distribution, thereby obtaining the centroid position of the target. In order to track the object, the window size needed to be adjusted until the search area was at the center of mass.

In summary, the flow of the Camshift algorithm includes three parts: obtaining the color histogram, calculating the Meanshift algorithm, and setting the initial value of the next frame of image sequence.

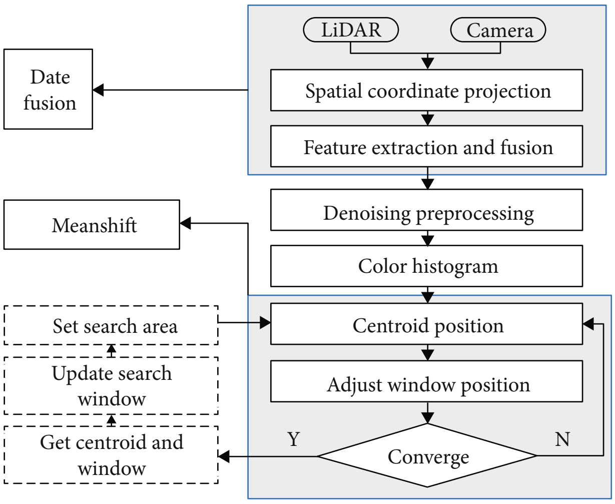


Figure 2: Camshift algorithm information fusion control flow chart.

3.1. Get the Color Histogram.

Due to the changes in lighting brightness in the interactive environment, the RGB color space is very susceptible to interference. Compared with the RGB color space, the HSV color space has a more stable performance on the changes in lighting brightness, so we change and convert the color space of the image accordingly.

Formula (4) is used to calculate the H component in the pixel and obtain the color histogram.

$$p_u(y) = \sum_{i=1}^{n} \delta[b(x_i) - u], \tag{4}$$

where δ represents the statistical function, and i is the feature value of the image, $b(x)$ is the number of pixels with characteristic value i in the image, and u represents normalization.

Finally, the color probability replaces the pixel value of the image. In a back-projected grayscale image, the larger the pixel value, the larger the original H value in the original image, and the higher the probability of the target pixel.

3.2. Meanshift Algorithm. Set the pixel in the search area as (x, y) and the pixel value as $I_c(x, y)$. The algorithm process is as follows.

Step 1. Initialize the search window.

Step 2. Calculate the centroid position of the fusion.

Zero moment:

$$M_{00} = \sum_x \sum_y I_c(x, y). \tag{5}$$

First moment:

$$\begin{cases} M_{10} = \sum_x \sum_y x I_c(x, y), \\ M_{01} = \sum_x \sum_y y I_c(x, y). \end{cases} \tag{6}$$

Centroid (x_c, y_c) of the search window:

$$\begin{cases} x_c = \dfrac{M_{10}}{M_{00}}, \\ y_c = \dfrac{M_{01}}{M_{00}}. \end{cases} \tag{7}$$

Step 3. Adjust and set the size of the search window to s and the length to $1.2s$.

$$s = \sqrt{\frac{M_{00}}{256}}. \tag{8}$$

First, we move the setting of the search window position to the centroid and use the value of M_{00} to adjust the size of the search window. If the window moving distance is greater than the set threshold, we repeat the above steps 2 and 3. When the distance between the center position of the search box and the center of mass is less than the set threshold, we perform a new round of target search on the next frame of image.

3.3. Set the Initial Value of the Next Frame of Image Sequence. Meanshift algorithm is the basis of Camshift algorithm, and the specific steps are as follows: first, the Meanshift operation was performed on each frame of image; second, the window information result of the previous frame was used as the initial value of the average shift operation of the next frame; finally, the above settings were repeated. The above steps constitute the Camshift algorithm.

However, the Camshift fusion tracking algorithm is very suitable for target tracking in a stable environment. When the background becomes complex, such as large-scale interference of similar colors or the case that the target is severely occluded, the algorithm is easy to lose the target. The reason is that this method is only based on the color space model and does not consider other characteristics of the target (such as the direction of movement). In response to the above problems, we consider incorporating motion estimation to ensure the stability of target tracking.

4. Multi-Information Fusion Algorithm Based on Kalman Filter

When there is occlusion in the scene, the tracking algorithm based on Camshift fusion is easy to lose the target. In order to solve the above problems, we proposed an improved tracking algorithm with Kalman motion estimation in this section. The algorithm used the Bhattacharyya distance to measure the degree of occlusion of the target, and the dynamic factor can vary with the interactive environment changes, so the tracking result of linear fitting can maintain a certain degree of robustness.

As shown in Figure 3, the content contained in the blue dashed box is the flow chart of the algorithm we proposed. It can be seen that the final output result of the algorithm is a linear fit of the adaptive mean shift algorithm and Kalman prediction results. Different weighting factors were adjusted according to the Bhattacharyya distance, so that even if the target of the interactive system is blocked, it can be maintained the function of predictive tracking. The improved combination algorithm in this paper was shown in Figure 4.

The calculation formula of the fitting result is shown as below.

$$X_{k+1} = \alpha \bar{X}_{k+1} + (1 - \alpha) Y. \tag{9}$$

Among them, X_{k+1} is the tracking position of the target at time $k + 1$, $\bar{X}_{k+1}$ is the predicted position of the Kalman filter, and Y is the optimal position obtained by the Camshift algorithm. The two results are linearly fitted through the adjustment of the weight coefficient α. The method we proposed combines the advantages of the KF and Camshift algorithm.

As you know, Bhattacharyya distance was generally used to measure the correlation between histograms. In this paper, we introduced Bhattacharyya distance to measure the degree of occlusion of the target. Before calculating the Bhattacharyya distance, we first needed to calculate the coefficient. The Bhattacharyya coefficient is as follows.

$$BC(p, q) = p(p(y), q) = \sum_{u=1}^{m} \sqrt{p_u(y) q_u}. \tag{10}$$

Among them, $P_u(y)$ is the color histogram of the target model, and q_u is the color histogram of the current tracking target subimage. According to the coefficient, the

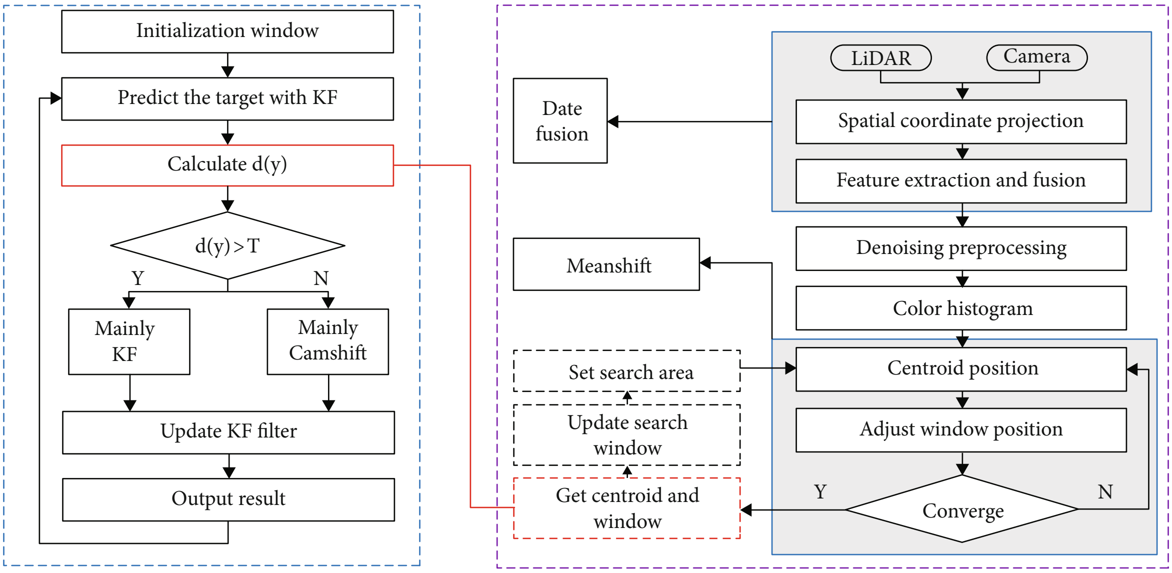


Figure 3: Flow chart of Camshift information fusion tracking algorithm based on Kalman filter.

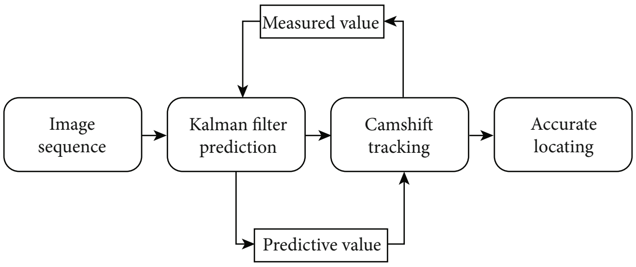


Figure 4: Camshift tracking algorithm improved by Kalman filter.

Bhattacharyya distance can be calculated, which is as follows:

$$d(y) = \sqrt{1 - BC(p, q)} = \sqrt{1 - \sum_{u=1}^{m} \sqrt{p_u(y) q_u}}. \qquad (11)$$

Based on experience, we set the threshold T as 0.85. If $d(y) < T$, the change of the interactive environment is not too serious, indicating that the tracking target is less occluded. If $d(y) \geq T$, a large deviation change occurs in the interactive environment, indicating that the tracking target is more blocked.

Finally, the algorithm steps are summarized as follows. First, the LiDAR and vision camera are fused, and the fused image is processed for noise reduction; second, the feature extraction and gesture detection are performed, and the target position is calculated according to the adaptive mean shift algorithm. Finally, the position is used as the initial information for Kalman filter. Specific steps are as follows: the distance $d(y)$ and the threshold T are compared, if $d(y) < T$, a smaller weight coefficient α is selected to linearly fit the Kalman filter prediction result and the Camshift algorithm calculation result; otherwise, a larger weight coefficient α is selected, which means the prediction result of Kalman filtering has a larger weight. Finally, the optimal position is obtained and updated to Kalman filtering. Through the above steps, the optimal estimation and prediction of target motion under complex interference background (such as occlusion) are realized.

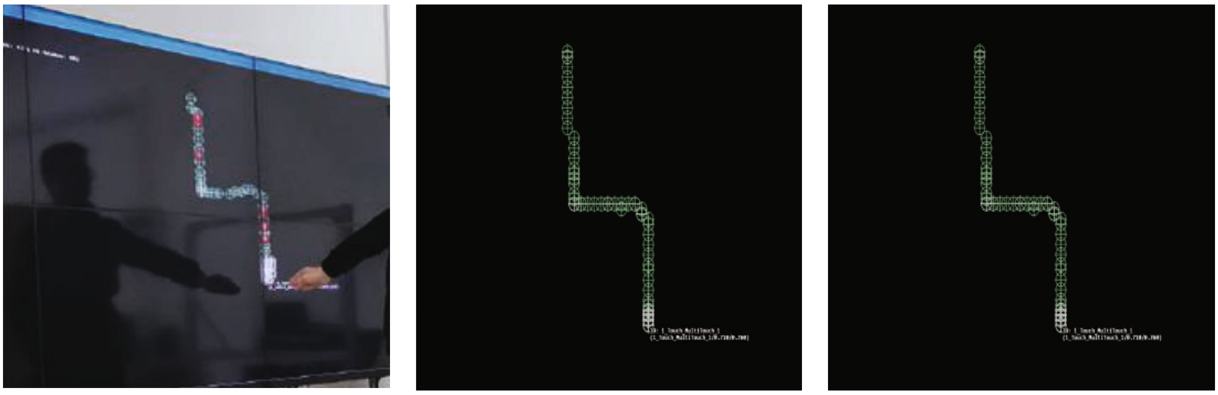
5. Experimental Results

This paper analyzed the two sets of experiments of gesture trajectory tracking and moving target tracking. In addition, we also verified the tracking characteristics of the system, the accuracy of interactive point positioning, the gesture recognition rate, and the tracking time.

5.1. Experiment 1: Gesture Trajectory Tracking Experiment. This article compared and analyzed the proposed tracking algorithm on the gesture tracking trajectory. After the trajectory curve passed the information fusion and filtering algorithm of the control host, the third-party interactive software Ventuz finally displayed the motion trajectory curve on the splicing large screen. In Figure 5, we visualized the tracking effects of the three algorithms.

Compared with the traditional adaptive drift algorithm and the original algorithm of the system, the tracking algorithm we proposed has obvious advantages, which can be seen from the results of experiment 1. The results showed that the tracking trajectory is continuous, accurate, and less interference.

5.2. Experiment 2: Positioning and Tracking Experiment of Moving Target. In order to further verify the algorithm, we continued to study the problems of occlusion and disappearance of interactive targets in complex environments. Based on the VC2016 compiler and OpenCV vision function library, the video image is collected by the CCD camera in real-time, and the position and movement information captured by the LiDAR sensor is fused to realize the positioning and tracking of the moving target. In order to compare the



(a) Original algorithm renderings

(b) Traditional Camshift renderings

(c) Algorithm renderings we proposed

Figure 5: System gesture trajectory tracking effect diagram.

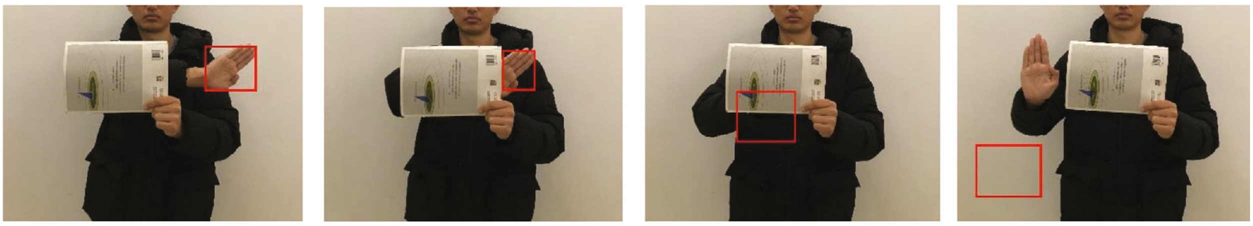


Figure 6: Experimental results of Camshift algorithm without Kalman filtering.

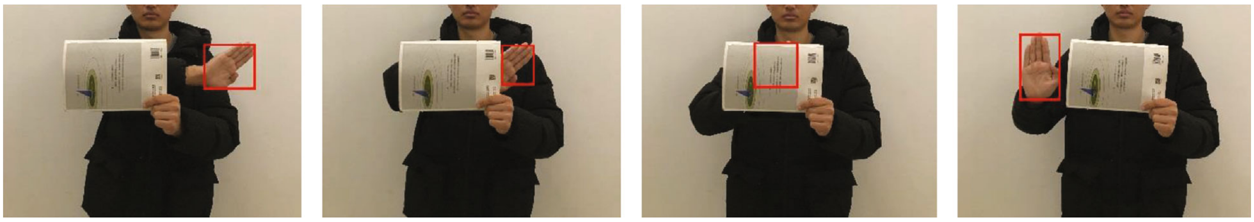


Figure 7: The experimental results of the algorithm in this paper.

tracking effect, the red box is used for marking in Figures 6 and 7. The specific experimental process is as follows.

(1) *Comparison of Book Occlusion Tracking Effect*. Starting from the 25th frame, recording was performed every 25 frames. When it reached the 100th frame, four different states of the experimental tracking were captured. The sequence contained four states: from far to near, partially covered, completely covered, and appear completely. To ensure objectivity, we conducted a total of three repeated experiments. In Figure 6, the results of Camshift algorithm without Kalman filtering were shown. For comparison, in Figure 7, the results of the algorithm proposed in this paper were shown

In Figure 6, at frame 25, the experimenter's hand approached the book from a distance, and the search red box could keep accurate tracking. At frame 50, the experimenter's hand was partially obscured by the book, and the search box still had no offset but was obviously smaller. At frame 75, the experimenter's hand was completely covered by the book. At this time, the red search box was obviously larger and tried to find the target. At frame 100, the experimenter's hand reappeared, but we can no longer locate and track. Experimental results showed that when the tracking target was occluded, the Camshift algorithm without Kalman filtering was not robust enough.

As a comparison, we also verified the algorithm proposed in this article, as shown in Figure 7. At 25 frames and 50 frames, the experimenter's hand could be accurately tracked; at 75 frames and 100 frames, the results were obviously different, though the experimenter's hands are completely covered by the book, but the red search box was kept in a relatively close position without too much shift. When the experimenter's hand appeared from behind the book, the search box quickly located the target and kept tracking, and the result showed that the tracking was successful.

Since the benchmark algorithm is the Camshift algorithm, which was based on a single color histogram model, the target was suitable for target tracking in a stable environment. When the target was severely occluded (for example, in Figure 6, the hand was completely blocked by the book

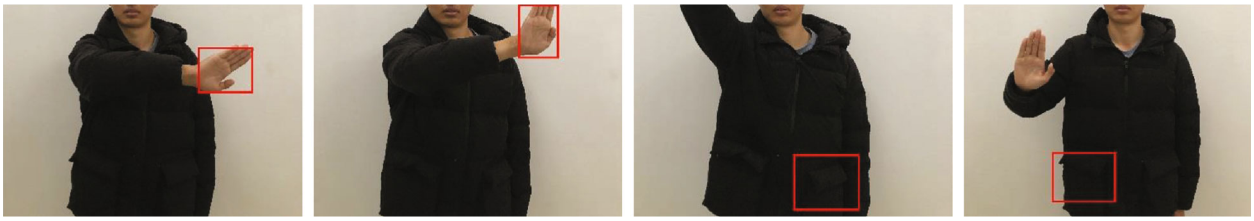


Figure 8: Experimental results of Camshift algorithm without Kalman filtering.

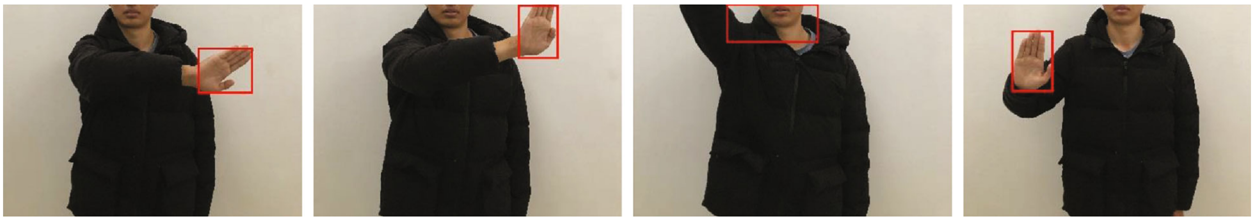


Figure 9: The experimental results of the algorithm in this paper.

at frame 75), the algorithm often could not adapt to this change, resulting in the loss of tracking the target. The algorithm proposed in this paper added Kalman filtering, which integrated the motion estimation of the tracking target. The Bhattacharyya distance was used to measure the degree of target occlusion, and the dynamic factor that changed with the environment was used for linear fitting to ensure the robustness of the tracking result.

(2) *The Experiment in which the Object Disappeared and Reappeared*. The experimenter made a circular motion by hand, disappeared over the effective interactive area, and finally reappeared in the effective area. To ensure objectivity, three repeated experiments were carried out. The following was a set of representative experimental results, which were shown in Figures 8 and 9

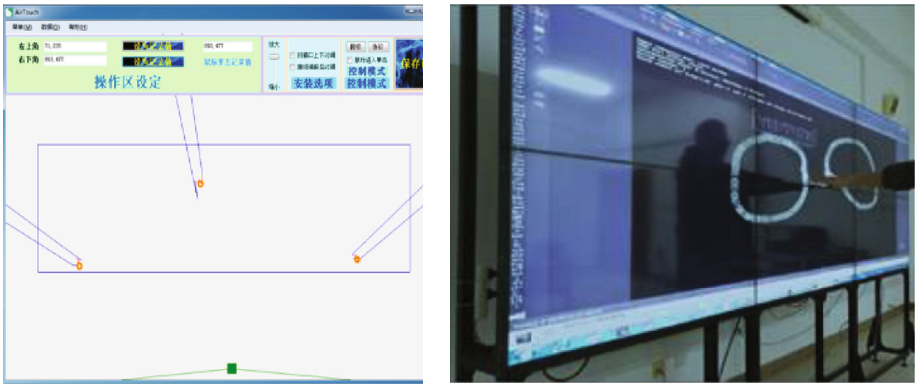
In Figure 8, start from the 20th frame, we recorded the tracking image every 30 frames and ended at 110 frames. In Figures 8 and 9, we first compared the 20th frame and the 50th frame. When the experimenter's hand always stayed in the effective interactive area, the above algorithm could keep accurate tracking of the target. However, there was a difference between frame 80 and frame 110. At frame 80 and frame 110 of Figure 8, the experimenter's hand disappeared from the effective interaction area. At this time, the search box is obviously shifted, and when the experimenter's hand reappeared in the effective interactive area, the search box could not be located on the position of the user's hand, and the results showed that the tracking was not success. At frame 80 and 110 in Figure 9, even if the experimenter's hand disappeared in the effective interactive area, the red search box did not deviate too far and remained in place. When the experimenter's hand reappeared in the area close to the target, our proposed algorithm could quickly find and track.

Combining the results of experiments 1 and 2, it could be concluded that the improved tracking algorithm in this paper had a good tracking performance on moving targets, especially when it encountered complex changes such as occlusion or disappearance. Since the tracking algorithm we proposed combined the Kalman filter, it proved that this algorithm effectively solved the problems of interactive point drift and target tracking failure.

According to the requirements of the interactive system, we performed specific functional tests and user evaluation scoring tests on the interactive system. Through these tests, it can be seen that the previous work such as noise reduction, calibration, tracking, and information fusion was very effective. The interactive effect demonstration effect was shown in Figure 10.

5.3. Tracking Algorithm Performance Comparison. For target tracking, accuracy is the most basic evaluation index. In order to obtain the recognition accuracy of gesture actions, we compared the Camshift algorithm, KF algorithm, and the improved algorithm we proposed, as shown in Table 1. In the experiment, we selected a total of five gestures, mainly including moving the hand up, moving the hand down, moving the hand to the left, moving the hand to the right, and doing arc motions. Each experiment was repeated 100 times.

From the analysis of a single indicator, compared to the Camshift and the KF algorithms, when the hand moves up, the recognition rate of ours reaches 90%, an increase of 7% and 2%, respectively; When the hand moves down, the recognition rate of ours reaches 91%, an increase of 6% and 4%, respectively. When the hand moves left, the recognition rate of ours reaches 94%, an increase of 13% and 4%, respectively. when the hand moves right, the recognition rate of ours reaches 93%, an increase of 16% and 3%, respectively. And when the hand makes arc movement, the recognition rate of ours reaches 92%, an increase of 14% and 1%, respectively. In summary, the average recognition success rate of ours can reach 92.0%, which is a significant improvement compared to other algorithms. The reason for a small



(a)

(b)

Figure 10: Interactive effect demonstration ((a) multitouch diagram; (b) gesture tracking effect diagram).

Table 1: Gesture recognition success rate comparison experiment.

Algorithm	Move up	Move down	Move left	Move right	Arc movement	Average value
Camshift	83%	85%	81%	77%	78%	80.8%
Kalman filter	88%	87%	90%	90%	91%	89.5%
Ours	90%	91%	94%	93%	92%	92.0%

Table 2: Occlusion comparison experiment results.

Algorithm	No occlusion	Partial occlusion	Severe occlusion	Tracking time
Camshift	83%	75%	54%	28 ms
Kalman filter	88%	82%	69%	53 ms
Ours	91%	89%	80%	62 ms

number of inaccurate recognition results may be related to the inconsistency of the frame rate of the fused image, or it may be related to the number of templates in the experiment.

In order to further verify the effectiveness of the algorithm proposed in the paper, we repeated the occlusion experiment described above several times and added tracking time as an evaluation indicator, as shown in Table 2. When there is no occlusion, the tracking effect of the three methods is better, and the success rate is above 80%. When partial occlusion occurs, the success rate of the three methods has decreased to varying degrees, but ours can still reach 89%. As the occlusion becomes severe, the success rate of the Camshift algorithm is only 54%, so the tracking target is often lost, while the KF algorithm is relatively good, which can reach 69%. It is worth mentioning that our algorithm has the highest tracking success rate, which can reach 80%. Compared with the previous two algorithms, the tracking time of ours is increased to 62 ms, but it still has a high real-time performance.

In summary, the experimental results show that the improved tracking algorithm based on Camshift in this paper further improves the overall interactive accuracy of the interactive system under different environmental conditions and has a relatively good large-screen interactive effect. Furthermore, the interactive system based on the combination of LiDAR and camera has high practical value.

5.4. The Actual Test of the Interactive System. In order to understand the user's actual experience of the system, we invited 30 students to participate. The system's scoring indicators mainly included system total score, command recognition, and interactive content. The students evaluated the system from four aspects: image fluency, recognition accuracy, real-time performance, and interactive fun. The highest score for each aspect was 100, and the lowest score was 1. We scored each content of 30 students and took the average and summary methods for statistics, as shown in Figure 11.

In Figure 11, the evaluation content scores corresponding to each dimension were all above 70 points, indicating that the user had a good overall experience of the system. However, the feedback real-time score of the system was slightly lower, so we could also optimize the algorithm to reduce the amount of calculation and shorten the feedback time. The experience scoring results of the students showed that some aspects of the system still needed to be improved.

According to the operating results of all the above experimental tests and analysis systems, the following conclusions can be drawn:

(1) The system interaction point is accurate. Even if the user moves beyond the effective LiDAR emission area and then returns to the effective area again, the positioning point remains at a relatively accurate position without interaction drift and inaccurate positioning

(2) The system runs stably, the interactive image is clear and stable, and the improved denoising algorithm

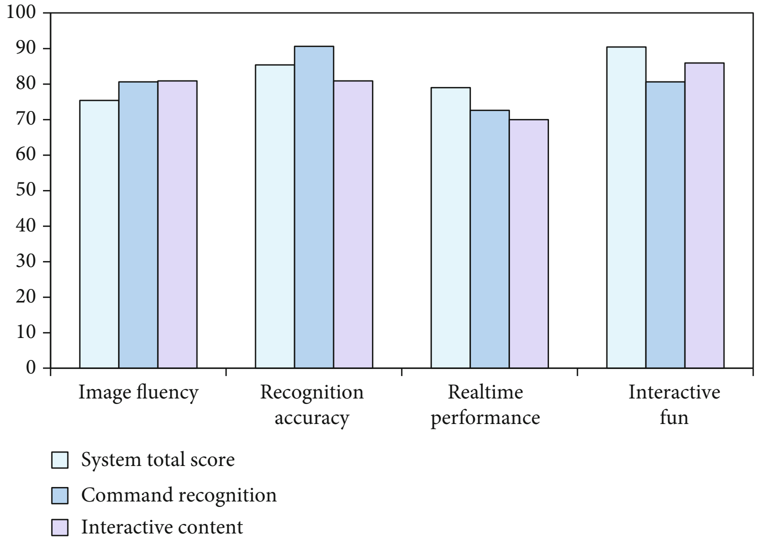


Figure 11: Interactive system user rating chart.

retains the details and depth information such as brightness and texture. Even if the external environment changes after the denoising algorithm processing, the system can still operate stably and achieve better results

(3) The operation result may be disturbed. If the distance between the display screen and the LiDAR emitting surface is too close, the image detection will be affected, and it is difficult for the system to keep stable tracking of the target. Therefore, the installation position of the LiDAR and vision camera should be selected in a suitable location. Besides, the interactive system must be tested and adjusted in the early stage, so we can ensure that the appropriate interactive effective area is used

(4) The system has a rapid response speed. The time from data signal collection to command output is less than 50 ms, which can meet the real-time requirements of the interactive system. The sensor collection information is basically consistent with the projection scene, which combines image denoising and target tracking. Technical improvements have effectively improved the performance of the interactive system and can maintain good robustness in the face of complex and changing interactive environments

6. Conclusion

This paper proposed a gesture detection fusion method based on LiDAR sensors. Furthermore, we introduced the Bhattacharyya distance to measure the degree of occlusion of the target and proposed an improved Camshift tracking algorithm based on Kalman filtering. Experimental results prove that the algorithm proposed in this paper can effectively solve the problem of interactive point positioning and tracking. When the interactive point exceeds the effective interactive area, the real-time predictive position of the algorithm can still keep accurate tracking, which can effectively solve the robustness problem of target tracking in interactive systems. Our future work focuses on extending the tracking target and considering the introduction of visual attention mechanisms to increase the speed of tracking.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The work is supported by the National Natural Science Foundation of China (no. 61873176), the Natrual Science Foundation of Shandong Province (Grant No:ZR2017LF010), the Qing Lan Project of the Higher Education Institutions of Jiangsu Province ([2019] 3, [2020] 10, [2021] 11), Basic science research project of Nantong (JC2020154). The authors would like to thank the referees for their constructive comments.

References

[1] D. M. Lyons and D. F. Hsu, “Combining multiple scoring systems for target tracking using rank-score characteristics,” *Information Fusion*, vol. 10, no. 2, pp. 124–136, 2009.

[2] B. Kim and D. Park, “Novel target segmentation and tracking based on fuzzy membership distribution for vision-based target tracking system,” *Image and Vision Computing*, vol. 24, no. 12, pp. 1319–1331, 2006.

[3] S. Kim and H. Choi, “Convolutional neural network for monocular vision-based multi-target tracking,” *International Journal of Control, Automation, and Systems*, vol. 17, no. 9, pp. 2284–2296, 2019.

[4] C. Pei, J. Zhang, X. Wang, and Q. Zhang, “Research of a nonlinearity control algorithm for UAV target tracking based on fuzzy logic systems,” *Microsystem Technologies*, vol. 24, no. 5, pp. 2237–2252, 2018.

[5] G. Zhao, S. Xu, and Y. Bo, “LiDAR-based non-cooperative tumbling spacecraft pose tracking by fusing depth maps and point clouds,” *Sensors*, vol. 18, no. 10, p. 3432, 2018.

[6] R. Opromolla, G. Fasano, G. Rufino, and M. Grassi, “A review of cooperative and uncooperative spacecraft pose determination techniques for close-proximity operations,” *Progress in Aerospace Sciences*, vol. 93, pp. 53–72, 2017.

[7] J. Boger-Lombard and O. Katz, “Passive optical time-of-flight for non line-of-sight localization,” *Nature Communications*, vol. 10, no. 1, pp. 3343–3349, 2019.

[8] L. Huang, S. Chen, J. Zhang, B. Cheng, and M. Liu, “Real-time motion tracking for indoor moving sphere objects with a LiDAR sensor,” *Sensors*, vol. 17, no. 9, p. 1932, 2017.

[9] L. Yu, J. Huang, and S. Fei, “Robust switching control of the direct-drive servo control systems based on disturbance observer for switching gain reduction,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 66, no. 8, pp. 1366–1370, 2019.

[10] X. Qin, G. Wu, J. Lei, F. Fan, X. Ye, and Q. Mei, “A novel method of autonomous inspection for transmission line based on cable inspection robot LiDAR data,” *Sensors (Basel, Switzerland)*, vol. 18, no. 2, p. 596, 2018.

[11] G. Nützi, S. Weiss, D. Scaramuzza, and R. Siegwart, “Fusion of IMU and vision for absolute scale estimation in monocular SLAM,” *Journal of Intelligent & Robotic Systems*, vol. 61, no. 1-4, pp. 287–299, 2011.

[12] L. Yu, C. Li, and S. Fei, “Any-wall touch control system with switching filter based on 3D sensor,” *IEEE Sensors Journal*, vol. 18, no. 11, pp. 4697–4703, 2018.

[13] S. Gehrig, N. Schneider, R. Stalder, and U. Franke, “Stereo vision during adverse weather – using priors to increase robustness in real-time stereo vision,” *Image and Vision Computing*, vol. 68, pp. 28–39, 2017.

[14] G. Dai, L. Yu, H. Xu, Z. Hou, and S. Fei, “Federated filter algorithm with positioning technique based on 3D sensor,” *Circuits Systems & Signal Processing*, vol. 37, no. 6, pp. 2613–2628, 2018.

[15] R. Fan and N. Dahnoun, “Real-time stereo vision-based lane detection system.,” *Measurement Science & Technology*, vol. 29, no. 7, p. 074005, 2018.

[16] J. J. Sofonia, S. Phinn, C. Roelfsema, F. Kendoul, and Y. Rist, “Modelling the effects of fundamental UAV flight parameters on LiDAR point clouds to facilitate objectives-based planning,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 149, pp. 105–118, 2019.

[17] C. Debeunne and D. Vivet, “A review of visual-LiDAR fusion based simultaneous localization and mapping,” *Sensors*, vol. 20, no. 7, p. 2068, 2020.

[18] S. Royo and M. Ballesta-Garcia, “An overview of LiDAR imaging systems for autonomous vehicles,” *Applied Sciences*, vol. 9, no. 19, p. 4093, 2019.

[19] A. Broggi, P. Cerri, S. Ghidoni, P. Grisleri, and Ho Gi Jung, “A new approach to urban pedestrian detection for automatic braking,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 4, pp. 594–605, 2009.

[20] A. Petrovskaya and S. Thrun, “Model based vehicle detection and tracking for autonomous urban driving,” *Autonomous Robots*, vol. 26, no. 2-3, pp. 123–139, 2009.

[21] D. Streller and K. Dietmayer, “Object tracking and classification using a multiple hypothesis approach,” *IEEE Intelligent Vehicles Symposium*, pp. 808–812, 2004.

[22] Q. Guo, F. Wu, S. Pang et al., “Crop 3D—a LiDAR based platform for 3D high-throughput crop phenotyping,” *Science China Life Sciences*, vol. 61, no. 3, pp. 328–339, 2018.

[23] R. Rossi, J. Ciparisse, A. Malizia, M. Gelfusa, and P. Gaudio, “Multiwavelength differential absorption LiDAR to improve measurement accuracy: test with ammonia over a traffic area,” *Applied Physics B*, vol. 124, no. 7, pp. 1–14, 2018.

[24] Z. Chen, R. Fan, X. Li et al., “Accuracy improvement of imaging LiDAR based on time-correlated single-photon counting using three laser beams,” *Optics Communications*, vol. 429, pp. 175–179, 2018.

[25] Z. Luo, M. V. Mohrenschildt, and S. Habibi, “A probability occupancy grid based approach for real-time LiDAR ground segmentation,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 3, pp. 998–1010, 2020.

[26] Y. Lyu, L. Bai, and X. Huang, “ChipNet: real-time LiDAR processing for drivable region segmentation on an FPGA,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 5, pp. 1769–1779, 2019.