Tracking and Locating Moving Targets Based on Kinect

Xusha¹, a, Dengjiahao², b, Duanzuodong³, c and Fanqiang⁴, d

¹Institute of electrical and mechanical, Beijing Institute of Technology, Beijing, China;
²Institute of electrical and mechanical, Target detection and damage control laboratory, Beijing Institute of Technology, Beijing, China;
³Institute of electrical and mechanical, Beijing Institute of Technology, Beijing, China;
⁴Institute of electrical and mechanical, Beijing Institute of Technology, Beijing, China;

abitxusha@163.com;bbitdjh@bit.edu.cn;duanzuodong@bit.edu.cn;dfanqiang@163.com;

Abstract. With the development of amusement robot, this paper is based on badminton robot, This paper propose a simple and extremely efficient tracking algorithm. Using device sensor can obtain the moving target three-dimensional coordinates, the algorithm is based on the depth of the moving target information, combining the background subtraction to remove some fixed background, the nonlinear cross filtering algorithm is proposed to remove noise, which establishes the rules for target recognition based on list sorted collection, to realize the moving target detection. According to motion model to forecast the object orientation of the next frame, combined with the a priori and real-time detection target state to the target location, at the same time by time lag compensation to further improve the real-time and accuracy. Through experiments, the results show that the prediction and actual error of the drop point are within the range of 30*30mm², which meets the research requirements of 200*200mm². Robot can return the ball with an 84% chance. The effectiveness of the proposed tracking and target algorithm is proved.

1. Introduction

At present, the tracking algorithm based on vision is becoming more and more mature. According to different tracking methods, the tracking algorithm can be divided into real-time detection and Combine prior and real-time data detection. Current mainstream tracking algorithms are based on prior and real-time data.

About moving target tracking algorithm, in the early 1960 s, kalman presents a new linear filtering and prediction theory of reason (kalman filter), its linear gaussian system tracking effect is very good, can do it without deviation estimates for moving target condition [1]. In 1975, Fukunaga first proposed the Mean shift tracking algorithm [2], which is based on kernel function estimation method. Therefore, there is no restriction on the movement form of the target and the type of noise. The Camshift algorithm is also proposed, which changes the window from fixed window to adaptive window[3]. In the 1990s, a particle filtering algorithm was proposed because it is based on the posterior probability to express the distribution of random state particles and can be used in any form of state space model[4]. However, the particle filter will increase with the increase of the number of particles, which is not suitable for real-time tracking. Then someone came up with the idea of separating the target from the background to achieve the target tracking, so there was a tracking algorithm based on the support vector machine(SVM)[5]. This algorithm is by finding a suitable surface cut the target and the
background, the algorithm is not affected by the influence of state space model of the target, but since it is done by classification learning divided the selection, so in a small sample, nonlinear, such as machine learning problems in performance is good, but the SVM algorithm for large-scale training sample is difficult to implement. In 2015, Girshick Microsoft research team led by successively put forward the Fast -R(CNN) [6] and Faster - R (CNN) [7] two deep learning model based on convolution neural network, improved efficiency, shorten the testing of CNN model available. Faster r-cnn has become a mainstream target detection method, but it cannot meet the real-time requirements in speed. Joseph Redmon et al. from the university of Washington proposed YOLO in 2016[8].2016 CVPR YOLO v2 and YOLO9000 [9], two model 2018 CVPR YOLOv3 model is put forward [10], YOLOv3 model than the previous model of complex, can change the size of the model structure to weigh the speed and precision, improves the precision, but need large samples need for training, not easy to achieve the goal of location.

In this paper, using of a new depth sensor Kinect[11], there is direct access to the depth of the moving target image, after filtering out noise, and then we propose a combining the background difference and the target sample sorting target detection method in order to realize the recognition of target, and then the actual conditions of badminton sports movement model is established, taking the two frame images of the target location, velocity parameters, generation into the model to estimate the placement. The proposed method can achieve target location and tracking easily, which can acquire good quickness and robustness.

2. Tracking algorithm for moving objects
The proposal tracking algorithm is put forward aimed at badminton robot, whose purpose is to hit the badminton back, to achieve this goal, there is necessary to calculate the coordinates of the landing site, the plane coordinates and the moment of hitting the ball in a certain height. According to the tracking system adopted in this paper, the information we can obtain directly is the depth image captured by the Kinect sensor. The general tracking algorithm is divided into three stages: motion target detection and target tracking and motion prediction.

2.1. Motion target detection
Considering limited vision field of Kinect vision, badminton can’t be continuous within the view of depth device camera, so target detection is regarded as an essential step, after that, to determine whether the object is within the scope of the device and whether the object is the badminton. Since the background of this paper is relatively simple, there is no target feature recognition like the general tracking algorithm. Instead, real-time detection is used to track the target.

Initial depth images include a lot of noise[12], firstly, filtering out primary noise and foreground and background, than combining the background subtraction to remove the inherent background. then to distinguish between man and badminton by the number of the target image patch, the depth image target image was left with a small amount of noise, this paper is to sort the rest of the patch, according to the rule that the distance between the sampling point and the median point is less than a certain value to resample, Considering the actual size of badminton in this paper, the critical value is 50mm.Finally, the average value of the target image at this time is used to describe the location of the target.

2.1.1. Depth image preprocessing and filtering the noise. In this paper, a cross - filtering algorithm based on depth information is proposed for smoothing and denoising depth images, as shown in figure 1.

The specific algorithm of cross filtering is:
- take 5 pixels as shown in figure 1. (where $depth_i$ represents the depth of point i)
- if $depth_i - depth_{i-1} \geq 60(i = 0,1,3,4)$, then center point 2 is filtered out.
- if $(depth_i + depth_{i-1} - 2 \times depth_2) \geq 10(i = 0,1)$, then center point 2 is filtered.
Comparing the two sensor filtering algorithms with cross filtering, as shown in the figure 2, the results show that cross filtering is more effective.

![Cross-filtering dot layout diagram](image)

**Figure 1.** Cross-filtering dot layout diagram.

![Image 2](image)

**Figure 2.** (a) the original grayscale image. (b) the image after mean filtering. (c) a gaussian filtered image. (d) a cross-filtered image.

2.1.2. Filtering the background. A background subtraction is to use the current detection image minus the setting background model, when the pixel difference value is greater than a certain critical value, then the pixels belongs to the moving target, the opposite is the background. In the static background, the adaptive effect is good and the real time is good. It is also applicable in dynamic background, but the background should be relatively simple, otherwise the establishment of background modeling will be difficult to accurately depict the real background situation. The core of background subtraction is the establishment of background model.

The filtering of the ball net is based on the current coordinates of the robot. The coordinate position of the net relative to Kinect can be calculated. Different filter strategies have different effects, the filtered part is shown in figure 3:

Vertical filtering is convenient to implement and has a small amount of calculation. However, if the opponent is hitting the ball in front of the net, it can detect the ball in a short time. As for oblique filtering, although this method improved the problem of short net time, in order to guarantee the network filtered clean, because the need according to the device in different network location adjustment filtering border, amount of calculation is bigger than the former method, so vertical filtering is more practical.

![Filtering boundary](image)

**Figure 3.** (a) vertical filtering, the lower left side of the filtering, boundary is the part that needs to be filtered out. (b) oblique filtering field and the ball in front of the net.
2.1.3. Extraction of target location. Considering that badminton’s mass is concentrated in the ball’s head, the Kinect detects different parts of the ball when it flies in different directions. To get a more accurate ball head position, try to filter the ball tail as much as possible. In the course of flight, the speed direction of badminton is basically the same as that of the ball head. If the detected point coordinates are set as \((x, y, z)\), the velocity of the ball \(v = (v_x, v_y, v_z)\), then each point is sorted according to the value \((x, y, z) \times (v_x, v_y, v_z)\), then we can obtain the position of the previous frame and the next position of each point in the badminton movement direction. The test point itself exists error, if the previous department as a ball head, it may be due to inaccurate measurement of some points, which will lead to the result that the detected coordinates are a few centimeters in front of the ball head and the actual ball head parts will be scrapped. Therefore, the first 3/10 of all points is selected as the ball head for improvement.

2.1.4. Time compensation. Because the sampling time of robot detection current coordinate and Kinect sampling time are controlled by two different clocks. When the two sampling times are out of sync, the badminton coordinates obtained by adding the two coordinates have certain deviation from the actual coordinates of badminton, the faster the robot moves, the greater the deviation. Therefore, time compensation is needed. Time compensation is built on the condition that we assume that the speed and angular velocity of the robot remain constant between two sampling moments.

Due to time lag between the computer from the robot sampling time to receive the robot coordinate is fixed generally, time lag from the sampling times to receive badminton relative coordinate is basic fixed, but two time difference is unknown, so the sampling time lag can be regarded as time lag between receiving two coordinates plus a fixed deviation, a fixed deviation is calculated by measured coordinates in stationary and movement states of the robot, when two coordinates are same, fixed deviation is the actual fixed deviation.

2.2 Simulation of motion trajectory and prediction of falling point of moving target.
According to relevant literature[13], the relationship between the air resistance and the speed of badminton is known \(f = \frac{1}{2} c D A \rho v^2\). The acceleration is \(a_1\), due to air resistance is opposite to the direction of velocity, \(a_1 = Fm^{-1} = k \frac{1}{2} |v| \sqrt{\rho} m^{-1} \frac{1}{2} |v| |v|^{-1} k |v| \sqrt{\rho}^{-1} v\). Due to gravity, \(a_2 = [0, 0, g] \frac{1}{2}\), combined with acceleration \(a = a_1 + a_2\). Iteration intervals is \(\Delta t\). If the current time velocity is \(v_k\), and the coordinate is \(w_k\), then the coordinates of the next frame are calculated as in equation (1), and the speed of the next frame is deduced as in equation (2).

\[
\begin{align*}
W_{k+1} &= W_k + V_k \Delta t \\
V_{k+1} &= V_k + a_1 \Delta t = V_k + (k \frac{1}{2} |v| |v|^{-1} V + [0, 0, g] \frac{1}{2}) \Delta t
\end{align*}
\]

Through experiments, the deviation between the predicted drop point and the actual drop point is compared with that of different parameters, and the parameter with the minimum deviation is taken. via adjusting the parameters \(k, k\), finally, \(k = 2.1\), \(k = 0.216\).

The badminton prediction algorithm through the detection of a badminton position point \(W_q = (x_q, v_x, z_q)\), and speed. Assuming that badminton in front and back frames move at uniform speed, the position coordinates of three consecutive frames are obtained as in equation (3). The velocity is shown in equation (4), and the direction of the velocity can be calculated according to the position of the three consecutive frames, as shown in equation (5).

\[
\begin{align*}
W_1 &= W_q - 33.3333V_q, W_2 &= W_q + 33.3333V_q, W_3 &= W_q - 33.3333V_q \\
V_1 &= 0.015 |W_3 - W_1| \\
\epsilon &= \tan((x3 - z1)(x3 - x1)^2 + (y3 - y1)^2)^{-1}
\end{align*}
\]
Where, the Angle (radian) $\theta_i$ between the velocity and the x-y plane; Then find the partial velocity $V_{x_i}$ in the x-y plane, and the partial velocity in the z direction is $V_{z_i}$, as shown in equation (6); And in the x-y plane, it's going to be on the X-axis $e_{2x}$, it's going to be on the Y-axis $e_{2y}$; The angular formula is shown in equations (7) (8).

$$
V_{x_i} = V_i * \cos(\theta_i), \quad V_{z_i} = V_i * \sin(\theta_i)
$$

$$
e_{2x} = (x_3 - x_1) \sqrt{(x_3 - x_1)^2 + (y_3 - y_1)^2}^{1/2}
$$

$$
e_{2y} = (y_3 - y_1) \sqrt{(x_3 - x_1)^2 + (y_3 - y_1)^2}^{1/2}
$$

To calculate the velocity scalar $V_i$ and direction quantity $\theta_i$ of the next frame according to the motion model, as shown in equations(1) and (2); Then the coordinate position of the moving target in the next frame is obtained as shown in equation (9).

$$
W_{i+1} = W_i + V_i \left[ \cos(\theta_i) * e_{2x}, \cos(\theta_i) * e_{2y}, \sin(\theta_i) \right]
$$

The trajectory is drawn according to equation (9) within a period of time. In this paper, it is set at 2 seconds, and 300 points can be calculated per second, so it is enough to draw the simulation trajectory.

When the height of the point in the track of badminton is less than or equal to the height of the hitting point, we record the plane coordinate position at this time. The coordinates of the hitting point $1hh_{xx}$. When the height of the point in the track of badminton reaches less than or equal to 0, the plane coordinate position at this time is recorded. That is the coordinates of the landing site $(,)_ground_{xx}$.

The weighted calculation of falling point coordinates and time of falling point. Since Kinect can detect many points during a badminton flight, the target position point of every three adjacent frames of images can be obtained. If the coordinates of the landing points sent to the robot each time are only obtained from the three newly detected points, it is easy to be affected by environmental factors (such as wind), and there will be a large error. Therefore, the previous calculation results are also taken into account. The closer the object is to the Kinect, the higher the detection accuracy. Take the weight value of the latest calculation result as 1/3, and the weight value of all the previous calculation results weighted is 2/3. The weight value of the latest calculation result is 1/2, and the weight value of all previous calculation results is 1/2. The weight of time is the same as the weight of the point of falling.

3. Experiments

In this section, two experiments were carried out to verify the validity of the algorithm, motion trajectory simulation experiment is mainly to prove the accuracy of the algorithm, batting experiment is designed to demonstrate the real-time performance and robustness of the algorithm.

3.1 Motion trajectory simulation experiment

The following experiment is to obtain the prediction of the drop point with the trajectory. As curves in the figure show, the blue star represents the point detected in real time, different adjacent three red curve points are put into the model to calculate the trajectory of badminton sport, and red circle stand for the weighted average of the shot point location, the blue star is the weighted average of the shot. The hitting point height is 0.8m. Its trajectory is shown in figure 4.
It can be seen that the weighted average can effectively reduce the impact of the points with large error and improve the accuracy of the predicted hitting points. And final position of the movement of the robot is determined by the last received point coordinates, finally, the shot point coordinates are shown in figure 5, the last ten hit points' coordinate error can achieve less than 30mm.

3.2 batting experiment
This experiment is done on the badminton robot in the standard badminton field, and lower boundary and left boundary are selected as the x axis and y axis, the lower left corner is regarded as the origin. The robot's center of mass calibration experiment is based on the field with a known location of the point by the center of mass of the corresponding method, the program is executed on panel, people in the opposite site will play badminton, badminton robot forecast placement, with the help of the control output added to the badminton robot, the badminton can hit ball back, and the probability of hit ball is high. Through 50 experiments, the statistics of hit times are shown in table 1. The moment the ball hits is shown in figure 6. This experiment mainly proves the validity and the prediction of hitting points.

According to the analysis of the experimental results, there are two types of badminton that can’t fight back, 1) badminton falls on the edge of the robot, which is mainly caused by the inaccuracy of the predicted hitting point, 2) the badminton falls under the base of the robot, which is mainly due to that the real-time performance of the algorithm is not up to the requirements and the robot is too late to respond.
4. Conclusion and Future work
Mobile target tracking is a hot research topic in computer vision, especially in mobile robot technology. Based on the visual tracking system of badminton mobile robot, this paper studies the tracking and positioning algorithm of moving target. For the noise problem of the acquired depth image, the traditional filtering algorithm has poor effect, and the proposed nonlinear "cross" filtering algorithm can effectively remove the noise point. The criterion of target recognition is put forward, to predict badminton sport based on the model, and in order to improve the precision, the combination of prior and real-time status is adopted to determine the final location, at the same time, the problem what the clocks are out of sync between sensor and control board is solved through time compensation, through verification on the robot platform, under the condition of rapid robot movement, rotation and arbitrary elevation Angle of Kinect, positioning accuracy is controlled within 3cm. The proposed algorithm can achieve a hit rate of 84%, proved the good real-time performance and accuracy of the proposed algorithm.

The algorithm proposed in this paper can't distinguish similar targets, if multiple targets appear at the same time, it just catch one. In addition, the background model is very complex and the applicability of the algorithm will be limited. Tracking algorithm based on window is vulnerable to keep out influence, and the real-time performance is bad, the recognition of multiple objective is also more complex, and end-to-end deep neural networks algorithm is a good way to solve the problem of real-time and environmental interference, and to achieve multiple targets recognition.

Acknowledgments
The research is supported by High-tech cross laboratory, we also would like to thank that the laboratory provides financial support and the experiment platform.

References

[1] Welch G, Bishop G. An Introduction to the Kalman Filter[M]. University of North Carolina at Chapel Hill, 2001.

[2] Yiyufeng, Gaoliqun, Guoli. random walk image segmentation algorithm based on Mean Shift [J]. Journal of Computer-Aided Design & Computer Graphics, 2011, 23(11):1875-1881.

[3] Exner D, Bruns E, Kurz D, et al. Fast and robust CAMShift tracking[C]// Computer Vision and Pattern Recognition Workshops. IEEE, 2010:9-16.

[4] Li P. Praticle filters based likelihood ratio approach to fault diagnosis in nonlinear stochastic systems[J]. IEEE Trans.on Systems Man. & Cybernetics Part C, 2001, 31:337-343.

[5] Keerthi S S, Shevade S K, Bhattacharyya C, et al. Improvements to Platt's SMO Algorithm for SVM Classifier Design[J]. Neural Computation, 2014, 13(3):637-649.

[6] Girshick R. Fast R-CNN[J]. Computer Science, 2015.

[7] Ren S, He K, Girshick R, et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks[J]. IEEE Trans Pattern Anal Mach Intell, 2017, 39(6):1137-1149.

[8] Redmon J, Divvala S, Girshick R, et al. You Only Look Once: Unified, Real-Time Object Detection[C]// IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, 2016:779-788.

[9] Redmon J, Farhadi A. YOLO9000: Better, Faster, Stronger[J]. 2016:6517-6525.

[10] Redmon J, Farhadi A. YOLOv3: An Incremental Improvement[J]. 2018.

[11] Wangzaoyong, Hebingwei. Research on indoor robot self-positioning based on Kinect sensor [J]. Machine Manufacture and Automation, 2014(5):154-157.

[12] Tanyan, Wangyujun, Lifeilong,Gediyu. Target tracking and 3d measurement based on Kinect [J]. Journal of southwest normal university (natural science edition), 2013, 09:101-105.

[13] Zhangjinhua. Study on the aerodynamics of badminton [D].Guangdong University of Technology, 2014.