Research Article

Design and implementation of badminton robot perception and control system

Liou Liu1, Liping Wang2,3 and Zhenwen Xu4

Abstract
With the rapid development of computer technology, target tracking has become an indispensable technology in the field of image processing. Outline-based matching algorithms are one of the most representative methods in the field of computer vision. The idea is to extract several characteristic vectors from the image and compares them with the characteristic vectors in the corresponding image template. The difference between the image and the template characteristic vector is calculated, and the category is determined by the minimum distance method. The badminton robot collects the depth image of the scene through the depth camera and then uses the machine vision theory to process the acquired depth image. To combine the image depth information to obtain the position of the badminton camera coordinate system in the three-dimensional space, the position of the site coordinate system is achieved. Finally, the position information of the badminton in the multi-frame images is used to predict the falling point of the badminton. The badminton positioning and the analysis of the falling point are completed. The badminton robot quickly runs to the predicted position of the badminton and completes a hitting task. To realize the high-speed continuous and smooth badminton action of the badminton robot manipulator, a new multi-objective manipulator trajectory optimization model is proposed. The experimental results show that the new trajectory optimization model can effectively reduce the energy consumption of the motor and improve the rotational efficiency, thus ensuring the response speed of the arm.

Keywords
Badminton robot, trajectory planning, shortest path algorithm, cubic spline interpolation

Date received: 16 December 2019; accepted: 4 February 2020

Topic: Robot Manipulation and Control
Topic Editor: Andrey V Savkin
Associate Editor: Bin He

Introduction
With the continuous maturity of robotics, robots engaged in ball games have gradually increased, and related research works such as table tennis robots and soccer robots have been relatively mature. However, the research and development of badminton robots is still in its initial stage, which is directly related to its design difficulty. First of all, the badminton robot’s batting action requirements are more complicated. To meet the competition requirements, the badminton robot can play a variety of simulated realistic games such as fixed-point ball, high throw ball, and ball picking. Second, badminton is not a regular sphere like...
football and table tennis. The sphere recognition is difficult, and the motion trajectory and its location are not easy to predict accurately. Due to the above reasons, the research and development requirements of badminton robots are very high, and there are few studies at home and abroad. Zhang et al.\textsuperscript{2} have developed the first robot that can be played against humans. The badminton is visually inspected using two off-site cameras and a computer system, and the robot’s vision system detects the position of the badminton from two perspectives so that it can obtain stereoscopic vision. However, since it can only move on straight slides, the catching area is limited and does not have a serving function. The portable badminton robot of the Dalian University of Technology hits the ball provided by the rotating device through a badminton racket.\textsuperscript{3}

The badminton robot has unique high-precision and fast motion mechanism, high-precision fast-moving target vision tracking algorithm, and robot real-time position correction technology to solve the extremely high requirements of badminton control and fast motion trajectory. Target tracking is an important branch of the computer vision field, which is the initial state (position, size, etc.) of a given target, and then estimates the new state of the measurement target from subsequent image sequences. There are many factors that affect the effects of tracking. The examples of factors are the changes in lighting, deformation of targets, and possible background interference. It should be said that there is currently no target tracking algorithm that can completely overcome these influencing factors. In this section, we will briefly describe the current target tracking algorithm. According to the algorithm, the target model can be divided into the overall representation method and the difference expression method. The general tracking algorithm usually obtains an expression model that characterizes the target object, and then finds the target area in the subsequent image through the template model. Patel and Bhatt\textsuperscript{4} proposed a typical overall expression tracking method that uses an expression model for off-line learning objectives. Bae and Yoon\textsuperscript{5} proposed a Gaussian mixture model, which uses the maximum expectation algorithm to adapt the algorithm to possible changes in the target surface in the actual tracking environment. The method proposed by Chen et al.\textsuperscript{6} obtains a series of online or off-line methods. A collection of target models is used to overcome surface changes in the target. Fang et al.\textsuperscript{7} further added a motion model based on the improved particle filter framework and combined the target multi-view expression model to track the target. In recent years, people have developed a target tracking algorithm based on sparse expression,\textsuperscript{8,9} which is roughly through the sparse expression of the target image, and then through L1-tracker to achieve effective target tracking.\textsuperscript{10} However, the fly in the ointment is that it involves a lot of complexity. The calculation also limits its application in real-time situations. It is proposed that a joint model of the expression of the target is to add the spatial configuration of the pixel and to estimate the local deformation degree of the target object. Therefore, the method can effectively overcome the deformation problem of the nonrigid body. The differential expression tracking algorithm simplifies the tracking problem into a binary classification problem, which is to select image blocks with target information from a large number of image block samples containing background. Dilawari et al.\textsuperscript{11} used the extended optical flow method to classify image blocks by support vector machine for target tracking. Gong et al.\textsuperscript{12} proposed a boost algorithm to select the features of the tracking. These are only using one positive sample and some negative samples to update the parameters of the classifier, and the classification mismatch caused by the noise in the actual process and the target surface change will lead to serious tracking drift problem. In view of the problem of tracking drift, Zheng et al.\textsuperscript{13} propose a semi-supervised boost method in which only the first frame of image is marked for updating the classifier, while the tracking instances in subsequent images are not used as marker samples. Researchers such as Li et al.\textsuperscript{14} have developed a tracking framework for multi-instance learning, which is studied by multiple positive and negative samples. Mercan et al.\textsuperscript{15} also developed a classifier for positive and negative samples, with the exception of adding structural constraints between sample templates. Recently, Liu et al.\textsuperscript{16} introduced an support vector machine (SVM) classifier based on a cyclic structure kernel matrix to track the targets. At present, the optimization objectives in the trajectory planning of the manipulator include mechanical energy consumption, trajectory smoothness, trajectory accuracy, shortest path, sudden change in velocity and acceleration, and exercise time.\textsuperscript{17,18} The current trajectory optimization research mainly focuses on the dual-objective planning based on the trajectory smoothing target combined with other optimization objectives. For example, researchers often use spline curves or polynomial interpolation to achieve the trajectory of the mechanical structure to improve the execution efficiency and trajectory smoothness of the manipulator. In addition, the trajectory is further optimized by combining intelligent algorithms. Costin et al.\textsuperscript{19} proposed an improved algorithm based on cubic spline interpolation, which effectively overcomes the sudden change of acceleration when the robot starts and ends and ensures the stability of motion. Kamble et al.\textsuperscript{20} used high-order genetic algorithms to achieve multi-objective optimization such as time, energy, and smoothness. For the stability requirements of the mechanical arm, Xiao et al.\textsuperscript{21} used a trajectory planning method based on cubic spline interpolation to generate a continuous smooth trajectory, but the acceleration variation is slightly larger, which makes the energy consumption of the mechanical arm increase. The abovementioned path planning algorithm optimization target mainly focuses on single target or dual target, and the multi-target requirement in the process of mechanical arm movement cannot be realized more effectively.
In practical applications, badminton tracking is difficult since badminton is an irregular object on the one hand. On the other hand, since badminton is in the air for a short time, it is necessary to ensure real-time performance. Under such a problem, this study proposes an outline-based template matching method for badminton, using the contour of the target object as the description model of the target object, and searching the target image contour in the contour map generated by the image to track the badminton in real time. The location and falling point of the badminton is predicted. In this study, the use of a few templates is possible to ensure real time, but also to achieve the largest possible matching accuracy, and image segmentation of images to improve the real-time badminton tracking. At the same time, after establishing the pose expression of the manipulator, the Newton downhill method is used to solve the nonlinear equations to obtain the inverse solution. Considering the optimization objectives such as optimal time, minimum energy consumption, and smoothness of motion trajectory, the shortest path algorithm is used to plan the optimal inverse solution, and the cubic corner motion trajectory, the shortest path algorithm is used to.

### Methodology

#### Outline-based template matching algorithm principle

First, we define the template of the target object as the vector \( d_i = (l_i, u_i)^T \), \( i = 1, ..., n \) associated with each point. The direction vector can be processed to obtain the image. The easiest way is to use the edge extraction method to get it. Extract the template of the target in the image, in addition to extracting the target object, it can also be packaged into the area of interest. Defining the coordinates as the relative coordinates of the center of gravity of the acquired region of interest is very convenient for subsequent calculations. We first represent the image of the search template as a form, and then find the vector \( e_{r,c} = (v_{r,c}, w_{r,c})^T \) at each point \((r, c)\) in the image. During the template matching process, the template must be compared to the various locations in the image. In general, this linear transformation is an affine transformation. It is useful to separate the translational part of the affine transformation. A linear transformation \( A \) can be derived from the point \( p_i^{00} = A p_i \) and the corresponding transformed direction vector \( d_i^T = (A^{-1})^T d_i \).

\[
\begin{pmatrix}
  a_{11} & a_{12} \\
  a_{21} & a_{22}
\end{pmatrix}
\]

Therefore, comparing the template to the image, the matching score must be unaffected by occlusion, confusion, and lighting changes. One way to accomplish this is to calculate the sum of the dot products of all points in the template and the dot product at the corresponding points in the image at a particular point in the image as a match score. This is the similarity measure of the point \( q = (r, c)^T \) at the template. This similarity is the translation part of the affine transformation. The formula for calculating the similarity measure is as follows

\[
s = \frac{1}{n} \left( \sum_{i=1}^{n} d_i^T e_{q+p} = \frac{1}{n} \sum_{i=1}^{n} \left( l_{i,r+c+1} + u_{i,r+c+1} \right) \right) \quad (2)
\]

If an edge filter is used to generate a template and the sum of the dot vectors of all points in the template and the dot product of the direction vectors in the image are calculated as a matching score in the search image, the similarity measure can be completely occluded and confusing. If the target object is partially occluded, the points at the corresponding positions on the template are not in the image, that is, the direction vectors of these points are small, and the overall effect is not substantially affected. Similarly, if there is a lot of noise in the image, the points corresponding to the template are equivalent to not being on the image, that is, the direction vectors of these points are small, which means that the noise does not affect the overall effect. However, the similarity metric formula provided in equation (2) does not exclude the noise interference of the illumination because the length of the direction vector extracted by the edge extraction algorithm is affected by external brightness. However, if a threshold is specified in a similar metric to limit the template, the similar metric needs to be within a range. The similarity measure in the formula below can satisfy this requirement

\[
s = \frac{1}{n} \sum_{i=1}^{n} \frac{d_i^T e_{q+p}}{||d_i^T|| ||e_{q+p}||} \quad (3)
\]

Since the direction vector is normalized, the similarity measure is not disturbed by the illumination noise because the modulo length of all direction vectors is one. This measure of similarity is also unaffected by occlusion and chaos, even if the noise causes a random direction vector, and its average effect on the sum is negligible. If the template is aligned with all direction vectors in the image, the similarity measure in equation (3) will find a very high score. If you use the edge extraction method to create a template and an image vector, the target object will not find the target object if the shading is reversed. However, in some cases, it is also necessary to search for the target in this case. This can be done in the following manner

\[
s = \frac{1}{n} \sum_{i=1}^{n} \frac{d_i^T e_{q+p}}{||d_i^T|| ||e_{q+p}||} \quad (4)
\]

The normalized similarity measure in equations (3) and (4) will return a score less than 1 as a suspected object. Under all template matches, a score of 1 means that the real
target is found. In this case, this score represents the occlusion in the template. For example, if 50\% of the badminton is occluded, the (average) score will not be greater than 0.5. The match score provides a meaningful value for the developer, and the developer can set the range of values to determine whether the target is found. The similarity metrics have good characteristics, and it is not necessary to fully find this similarity metric when the similarity metric calculated at the potential matching location reaches the developer-defined threshold $s_{\text{min}}$. We use the sum of the dot products that represent the $j$th element accumulated into the template.

Obviously, the sum of the dot products of all templates is less than or equal to 1.0. Therefore, if the matching score does not reach the required threshold $s_{\text{min}}$, the calculation can be stopped after the $j$th element when this condition $s_j < s_{\text{min}} - 1 + \frac{j}{n}$ is satisfied. This condition greatly increases the speed of the recognition process. To identify a template, you can create an image pyramid for the image and then search for the template in the image pyramid. The same filtering operation, such as edge filtering, is used when creating a template at each level of the image pyramid. This filtering operation will obtain the direction vector for each image point. Note that the image is not split, that is, no threshold segmentation or other operations are performed. This makes the algorithm truly immune to changes in illumination.

To determine the suspected target position, we need to search from the top of the image pyramid. That is, the image on the top of the image pyramid needs to calculate the template in all possible poses. The match score for a suspected target location must be greater than in order to end the meaningless lookup to speed up the search of $s_{\text{min}}$. Therefore, this seemingly powerful search strategy that requires a complete traversal of the entire image is actually quite efficient. After determining the suspected target location, we will search from the top of the image pyramid to the bottom and find the target at the bottom of the image pyramid. Although the position obtained by extrapolation has been very accurate for many applications, some applications are expected to yield more accurate results. This can be done by adjusting the least squares of the pose parameters. If an edge extraction algorithm is used, the direction vector of the template point is perpendicular to the object boundary, so the equation of the line is passing through the template point and tangential equation $t_i(r - r_i) + u_i(c - c_i) = 0$ to the object boundary. If an image point that represents the corresponding point to the template is used, we can minimize the following function to make the parameters more accurately

$$d(a) = \sum_{i=1}^{n} [t_i(r'_i(a) - r_i) + u_i(c'_i(a) - c_i)]^2$$  \hspace{1cm} (5)

It can be expected that the obtained badminton shape contour models are different in size, and some contours have a higher probability of matching to badminton than other contour models, if the selected badminton contour models have higher matching probabilities and can be in badminton depth. The sample set matching complements each other, so it is undoubtedly able to obtain a higher badminton detection rate and reduce the number of templates required to reach the corresponding badminton detection rate, thereby reducing the detection time of the badminton. After the badminton template is created, the badminton in the image can be detected to obtain the position of the badminton. The outline-based badminton template matching is to perform a point-by-point traversal of the zoomed and rotated badminton template in the target graphic. If the obtained matching value is higher than the set value, it is considered that the point is likely to be the badminton position. The image processing includes the gray transformation and binarization. Thus, the outline-based badminton template matching process is shown in Figure 1.

**Figure 1.** The outline-based badminton template matching process.

**Trajectory optimization based on cubic spline curve**

Using the D-H notation, a coordinate system is established for the manipulator, and a kinematic model is established based on a $4 \times 4$ homogeneous transformation matrix.
Badminton strokes can be roughly divided into positive and side shots. The badminton robot completes the action of hitting the badminton through two processes: the servo motor system provides power to move the robot arm along the predetermined trajectory to the target point, and finally the cylinder provides the hitting force to the racquet to achieve the striking action. This article mainly studies the process of the trajectory movement of the mechanical arm and optimizes the trajectory motion. To study the badminton robot to realize the high-speed continuous and smooth badminton action of the robot arm, the kinematic model is established by using the D-H model according to the following mechanical structure. According to the principle of D-H representation, the connecting rod coordinate system is established for the mechanical arm, as shown in Figure 2, where $h_0 = 155$ mm, $h_1 = 305$ mm, and $h_2 = 410$ mm. The three subcoordinates represent the three connecting rods.

Among them, the meaning of each member parameter is as follows: $\theta_i$ represents the joint angle of the $z_{i-1}$ axis (right-hand rule) from the $x_{i-1}$ axis to the $x_i$ axis, and $\alpha_i$ represents the $x_i$ axis (right-hand rule) from the $z_{i-1}$ axis to the $z_i$ axis. Deflection angle, $\alpha_i$, represents the offset distance from the intersection of the $z_{i-1}$ axis and the $x_i$ axis to the $i$th axis of the $i$th coordinate system, and $d_i$ represents the origin from the $i$-1 coordinate system to the $z_{i-1}$ axis and the $x_i$ axis. The distance of the intersection points along the $z_{i-1}$ axis, and $h_i$ represents the joint length of the $i+1$ joint. In the Cartesian coordinate system, in order to better study the position of a point, it is often necessary to convert the pose of the object to be described in different coordinate systems to another reference coordinate system. To describe the translational and rotational relationship between adjacent joints, the rotation transformation matrix $\text{Rot}(x, \theta)$ and the translation transformation matrix $\text{Trans}(a, b, c)$ are used to establish $4 \times 4$ for the coordinate system at each joint. The matrix is transformed homogeneously to represent its relationship to the previous coordinate system. By successive transformations, a transformed representation of the racquet coordinates relative to the pedestal coordinate system is obtained. According to the mechanical structure diagram and the D-H model of the mechanical arm linkage coordinate system, the composite coordinate transformation is performed on the mechanical arm. In different poses, $\theta_i (i = 1, 2, 3)$ changes accordingly. The following is the transformation matrix between the coordinate systems. $A_i^{i-1}$ represents the homogeneous transformation matrix of the $i$ coordinate system and the $i-1$ coordinate system, which is called the D-H transformation matrix of the adjacent coordinate system $i$ and $i-1$.

$$A_1^0 = \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 & 0 & 0 \\ \sin \theta_1 & \cos \theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(6)

$$A_2^1 = \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & 0 \\ \sin \theta_2 & \cos \theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(7)

$$A_3^2 = \begin{bmatrix} \cos \theta_3 & -\sin \theta_3 & 0 & 0 \\ \sin \theta_3 & \cos \theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(8)

The point $i^{-1}P$ is calculated as

$$i^{-1}P = A_i^{i-1} \times iP$$

(9)

The homogeneous transformation matrix $T$ of the $i$th coordinate system relative to the coordinate system of the frame is the continuous product of the homogeneous transformation matrices $A_i^{i-1}$. The equation form is as follows

$$T_i = A_1^0A_2^1A_3^2 \cdots A_i^{i-1} = \prod_{j=1}^{i} A_j^{j-1}$$

(10)

The transformation matrix of each link is multiplied to obtain a total homogeneous transformation matrix of the end point $P$ of the arm relative to the coordinate system of the base

$$T = A_1^0A_2^1A_3^2$$

(11)

$$^0P = T \times ^3P$$

(12)

Assuming that the coordinates of the center position $P$ of the robotic racket end effector are $P(x, y, z)$, the position of $P$ relative to the absolute coordinate system of the base is derived from the D-H transformation matrix and equations (6) to (9). The pose expression is
\begin{align}
x &= h_2 \cos \theta_1 \cos \theta_2 \cos \theta_3 - h_2 \sin \theta_1 \sin \theta_3 + h_1 \cos \theta_1 \sin \theta_2 \\
y &= h_2 \sin \theta_1 \cos \theta_2 \cos \theta_3 + h_2 \cos \theta_1 \sin \theta_3 + h_1 \sin \theta_1 \sin \theta_2 \\
z &= -h_2 \sin \theta_2 \cos \theta_3 + h_1 \cos \theta_2 + h_0
\end{align}

To satisfy the multi-objective optimization, the shortest path algorithm is introduced to solve the solution. The following defines each point \( P_i \) that needs to pass as a node on the shortest path, and the value of the corresponding adjacent node difference is \( \Delta P_i = P_{i+1} - P_i \) \((0 < i < n)\) (where \( \Delta P_i \) can be defined according to different demand, user-defined), then \( \Delta P_i \) also corresponds to the weight on node \( P_{i+1} \). For a trajectory, the initial position is given when selecting the next node provided that the next node has a solution. If there is no solution, then the node with the solution is taken. Therefore, the set of solutions with the smallest \( \Delta P_i \), which guarantees the change of the angle of rotation of each motor, is the smallest on a particular trajectory, \( \Delta P = \sum_{i=1}^{n-1} \Delta P_i \). The motor is controlled under the same conditions, and the optimized solution can satisfy the time requirement. The construction of a spline function using piecewise cubic Hermite interpolation is

\[\theta(t) = \sum_{i=0}^{n} \left[ y_i t \alpha_i(t)^3 + m_i \beta_i(t) \right]\]

According to this trajectory planning method, multi-objective optimization of the corresponding trajectory is realized for different badminton hitting actions.

**Results and discussion**

**Matching algorithm recognition effect**

Since the badminton adopts the outline-based template matching method and uses more matching templates, the badminton is directly detected in the image and the badminton detection time of the single image can reach more than 1 s, which is far from the requirement of real-time application. At the same time, it will cause a large number of mismatches, which increases the difficulty of subsequent processing. Therefore, it is necessary to process the image and extract a small portion of the image area containing the badminton. Since the value of each pixel of the depth map corresponds to the distance from the camera to the object, the entire image can read the distance of the object. Since the badminton and the surrounding objects have a long distance in the actual environment, in the depth map, the badminton area is an area where the block has a large difference in gray value from the surrounding background and has a special size and shape. The image is first subjected to threshold processing, and then the depth image is segmented using the region growing method. The basic idea of region growing is to group pixels with similar properties to form regions. Specifically, a seed pixel is first found as a starting pixel point, and then a pixel having a similar gray value with the pixel in the neighborhood around the pixel is merged into the region where the pixel is located. These newly found pixels are treated as new seed pixels to continue the above search process, and the search is performed until the imaginary lines with no similar gray values in the neighborhood. Thus, a candidate area is found: the pixel of the image which is less than or equal to 4 is a seed point and the pixel area is larger than 12. Some areas are excluded based on the pixel area greater than 12 and less than 1000. If \( L \) is the length of the outline and \( F \) is the area, \( C \) is defined as

\[C' = \frac{L^2}{4F\pi}\]

\[C = \max(1, C')\]

It is known that a circle shape factor \( C \) is 1.0. If the area is long or there is a hole in the area, \( C \) is greater than 1.0. The degree of compactness reflects the rudeness of the outline whether there is a hole in the area. Finally, when the gray level average of the detected area is greater than 12 and the gray value of center point in the area is greater than 30, the area density of the badminton is less than 1.6, and the area is between 12 and 900, the image is considered match to the real badminton. Therefore, the test obtains a large number of sample pictures from the camera and then randomly selects 613 effective images as test images from a large number of samples (see Table 1). The test picture has been removed from the picture without badminton, and the recognition rate of the badminton is tested on the program.

To test the actual hitting rate of the badminton robot, the tests are divided into 10 groups indoors, and 10 badminton were cast in each group. The hitting rate of each group is counted. The statistical results are shown in Figure 3.

As can be seen from the above table, the actual hit rate of the robot is 78%. Since the recognition rate of the badminton and the accuracy of the predicted position of the falling point are both above 90%, the actual hit rate of the robot is about 80%. In addition, due to the outline-based template matching algorithm, the human head and badminton rackets will cause interference in the recognition of badminton. Since the randomness of pitching, the robot cannot reach the actual landing prediction position.

Finally, it is necessary to continuously successfully...
identify the badminton location in a consecutive five images. It can be seen that although the actual badminton robot’s hitting rate is about 80%, there is still much room for improvement in badminton robots in terms of mismatch of badminton, bounce point prediction, and badminton robot safety.

**Robotic arm execution**

The multi-objective programming of the three-degree-of-freedom manipulator trajectory studied in this article is mainly based on the following robotic arm experimental platform for trajectory planning experiments. According to the actual situation of hitting the badminton, in the design of the mechanical arm structure, it is necessary to consider the multi-joint flexibility of the manipulator and also consider the rotational efficiency of the manipulator, so a three-degree-of-freedom mechanical structure is adopted. The bottommost turntable structure of the manipulator is used to simulate the human boom joint and has a large driving force. The second structure is rotated vertically to simulate the second joint of the person to achieve a wide coverage of the target point. The human wrist flexibly changes the position of the end effector for precise positioning. Finally, the badminton racket is fixed by the dead rod. The hitting force of the badminton racket is provided, and the cylinder enables the racket to obtain a larger speed and hitting force in a shorter time to complete the final striking action. The following mechanical structure can be used to better simulate the action of hitting the badminton. According to the trajectory optimization model introduced in the previous section, the trajectory planning flow chart is used to conduct experiments on the structure of the mechanical arm studied. It is assumed that the action of hitting the badminton is a side shot, as shown in Table 2.

According to the optimized inverse solution, the motor corner function $\theta(t)$ and its coefficient matrix are established by cubic spline interpolation

$$
\theta_1(t) = \begin{cases} 
M_{00}\varphi^3 + M_{01}\varphi^2 + M_{02}\varphi + M_{03}\psi; (0 < t < 0.1) \\
M_{10}\varphi^3 + M_{11}\varphi^2 + M_{12}\varphi + M_{13}\psi; (0.1 < t < 0.2) \\
M_{20}\varphi^3 + M_{21}\varphi^2 + M_{22}\varphi + M_{23}\psi; (0.2 < t < 0.3) \\
M_{30}\varphi^3 + M_{31}\varphi^2 + M_{32}\varphi + M_{33}\psi; (0.3 < t < 0.4) \\
M_{40}\varphi^3 + M_{41}\varphi^2 + M_{42}\varphi + M_{43}\psi; (0.4 < t < 0.5)
\end{cases}
$$

where $\varphi = t_{i+1} - t$ and $\psi = t - t_j$.

Furthermore, the manipulator uses the spline interpolation function to obtain the control parameters. Under the same time step, the control process is analyzed by the motor running to reflect the motor angle change, angular velocity change, and angular acceleration change. Substituting the motor corner function into the corresponding pose expression to obtain the trajectory point function, the trajectory curve is obtained by fitting. The angle, angular velocity, and acceleration variations of the solution and the general solution are optimized, as shown in Figures 4 to 6. By comparing the variations of each curve, it can be explained that the optimized solution satisfies the planning requirements of multiple targets such as continuous smoothing of the trajectory. Comparing the curves of the three motor angles, we can find that the angle change of the optimized solution curve is more stable than the general solution, and the final angle is smaller than the angle of the general solution. It can be explained that the optimal solution of the shortest path idea planning is used in the same time. The motor energy consumption and the motor angle change rate are significantly optimized. Comparing the three motor speed curves, it can be seen that the optimization solution has a smaller speed peak than the general solution, and the variation range is small. It can be explained that the shortest path planning can make the motor rotate more efficiently in the same time. Comparing the acceleration curves of the three motors, it can be seen that the peak of the acceleration of the optimized solution plan is smaller and the rate of change is smaller, which is better for the stability of the mechanical structure, the consumption of the motor is smaller, and the service life of the motor is ensured. For the same badminton striking action, at the same time, the general planning solution control process requires the motor to rotate at a larger angle, and its angular velocity and angular acceleration are larger.

Therefore, if the motor control is performed under the same conditions, the solution is optimized. The time required is better, and since the solution that satisfies the time is also selected from the feasible solution set (inverse solution set),

**Figure 3.** Results of badminton robot in actual batting tests.

**Table 2.** Locations of point $P$ in the end actuator.

| $P_i$ | 0 | 1 | 2 | 3 | 4 | 5 |
|-------|---|---|---|---|---|---|
| $x$ (cm) | 41 | 35 | 20 | 9 | 3 | −4 |
| $y$ (cm) | 0 | −2 | −7 | −9 | −21 | −28 |
| $z$ (cm) | 46 | 53 | 62 | 65 | 62 | 58 |
the integrated conditions can be used to meet the optimal time requirement. Further, the trajectories are compared using MATLAB. The point set $P_i$ of the end effector of the manipulator, ($0 < i < n$), obtains the line trajectory of the angle variations of (a) actuator 1 and (b) actuator 2.

![Figure 4. Angle variations of (a) actuator 1 and (b) actuator 2.](image)

The trajectory to obtain the general planning point $P$ motion trajectory is calculated according to the general solution data obtained by Newton’s downhill method using MATLAB.

The trajectory to obtain the general planning point $P$ motion trajectory is calculated according to the point $P$ according to the general solution data obtained by Newton’s downhill method using MATLAB.

![Figure 5. Angle velocity variations of (a) actuator 1 and (b) actuator 2.](image)

![Figure 6. Angle acceleration variations of (a) actuator 1 and (b) actuator 2.](image)
optimization solution obtained by the shortest path algorithm using MATLAB. The above trajectory planning curve is shown in Figure 7. The curve in Figure 7(a) is the line trajectory of the point set $P$ and the general planned point $P$ motion trajectory respectively. The curve in Figure 7(b) is the point set $P$ respectively. The point trajectory is the trajectory fold line formed by the point set $P$ through which the direct connection passes. If it is used as the trajectory of the mechanical arm, it will cause a large loss to the motor, and the curve is discontinuous, which can only be reflected to some extent. The trend of the trajectory curve and the trajectory cannot be used to realize the movement of the robot arm. The motion trajectory curve planned according to the general solution obtained by Newton’s downhill method. The curve satisfies the point $P$ that passes through each need, but the path is too long and the trajectory realization is more complicated. The motion curve fitted by the optimal solution obtained from the shortest path idea planning. Comparing the trajectory fold lines, it can be seen that the trend of the trajectory curve is consistent and the curve is continuously smooth. The curve path length of the optimized solution plan is significantly better than the general solution. In general, through this trajectory curve obtained by the planning algorithm, the multi-target requirement of the manipulator arm movement is satisfied, and the trajectory curve according to the optimization solution plan can be adopted.

Conclusion

This article completed the hardware and software implementation of the system. The badminton robot can capture the depth images of the scene through a depth camera and then use the machine vision theory to process the acquired depth images. By combining the image depth information, the position of the badminton camera coordinate system in the three-dimensional space is obtained. Then the position of the site coordinate system is converted, and finally the position information of the badminton in the multi-frame images is used to predict the falling point of the badminton. The badminton positioning and the analysis of the falling point are completed. The badminton robot quickly runs to the predicted position of the badminton and completes a hitting task. At the same time, for the multi-objective programming problem of the three-degree-of-freedom badminton robot manipulator trajectory, the shortest path algorithm is used to plan the optimal solution from the inverse solution and use the cubic spline function to establish the motor corner function to ensure the trajectory is continuous and smooth. The experimental results show that the trajectory planning algorithm effectively realizes the multi-objective planning of the bounce robot arm’s trajectory for the shortest time and smooth path, and also meets the need to reduce energy consumption during the motor rotation process to improve the rotation efficiency.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

ORCID iD

Liou Liu https://orcid.org/0000-0002-8902-4724

References

1. Depraetere B, Liu M, Pinte G, et al. Comparison of model-free and model-based methods for time optimal hit control of a badminton robot. Mechatronics 2014; 24(8): 1021–1030.
2. Zhang J, Ren Y, and Zhang D. Marrying tracking with ELM: a metric constraint guided multiple features fusion method. Pattern Recogn Lett 2019; 120: 82–88.
3. Pengsong L, Yu Z, and Davis K R. Research on the value of mental exercise from the “trust system” of sports in Chinese universities. Science 2016; 4(6): 489–493.
4. Patel M J and Bhatt B. A comparative study of object tracking techniques. Int J Innova Res Sci Eng Technol 2015; 4(3).
5. Bae S H and Yoon K J. Confidence-based data association and discriminative deep appearance learning for robust online multi-object tracking. *IEEE T Pattern Anal Mach Intellig* 2017; 40(3): 595–610.

6. Chen W, Liao T, Li Z, et al. Using FTOC to track shuttlecock for the badminton robot. *Neurocomputing* 2019; 334: 182–196.

7. Fang J, Xu H, Wang Q, et al. Online hash tracking with spatio-temporal saliency auxiliary. *Comput Vis Image Und* 2017; 160: 57–72.

8. Miao W, Li G, Sun Y, et al. Gesture recognition based on sparse representation. *Int J Wirel Mobile Comput* 2016; 11(4): 348–356.

9. Yang L, Zhao L, Bi G, et al. SAR ground moving target imaging algorithm based on parametric and dynamic sparse bayesian learning. *IEEE Trans Geosci Remote Sens* 2016; 54(4): 2254–2267.

10. Yang Y, Yang J, Liu L, et al. High-speed target tracking system based on a hierarchical parallel vision processor and gray-level LBP algorithm. *IEEE Trans Syst Man Cybern Syst* 2016; 47(6): 950–964.

11. Dilawari A, Khan M U G, ur Rehman Z, et al. Toward generating human-centered video annotations. *Circu Syst Signal Pr* 2019; 1–27.

12. Gong W, Zhang X, González J, et al. Human pose estimation from monocular images: a comprehensive survey. *Sensors* 2016; 16(12): 1966.

13. Zheng Q, Delingette H, and Ayache N. Explainable cardiac pathology classification on cine MRI with motion characterization by semi-supervised learning of apparent flow. *Med Image Anal* 2019; 3(1): 36–42.

14. Li Z, Liao B, Li Y, et al. Gene function prediction based on combining gene ontology hierarchy with multi-instance multi-label learning. *RSC Adv* 2018; 8(50): 28503–28509.

15. Mercan C, Aksoy S, Mercan E, et al. Multi-instance multi-label learning for multi-class classification of whole slide breast histopathology images. *IEEE Trans Med Imaging* 2017; 37(1): 316–325.

16. Liu M, Zhang J, Adeli E, et al. Landmark-based deep multi-instance learning for brain disease diagnosis. *Med Image Anal* 2018; 43: 157–168.

17. He C, Shao J, Zhang J, et al. Clustering-based multiple instance learning with multi-view feature. *Expert Syst Appl* 2019; 113027.

18. Wu Z, Zhang M, Wan S, et al. A new multiple instance learning algorithm based on instance-consistency. *Inter J Perform Eng* 2017; 13(4): 519–529.

19. Costin M, Vonică I, and Bivoli I. A priori cost of the energy to perform movement on a predefined path. *Acta Polytech Hung* 2018; 15(7): 223–241.

20. Kamble PR, Keskar AG, and Bhurchandi KM. Ball tracking in sports: a survey. *Artif Intellig Rev* 2019; 52(3): 1655–1705.

21. Xiao J, Feng Y, and Hu W. Predicting missing markers in human motion capture using L1-sparse representation. *Comput Animat Virt W* 2011; 22(2-3): 221–228.

22. Tan CH, Hou JH, and Chau LP. Motion capture data recovery using skeleton constrained singular value thresholding. *Visual Comput* 2015; 31(11): 1521–1532.

23. Bouachir W and Bilodeau G A. Exploiting structural constraints for visual object tracking. *Image Vision Comput* 2015; 43: 39–49.