Acta Technica *62* No. 3A/2017, 45–54

Research on tennis judgment system based on hawk-eye technology

JIAFU FENG[1]

Abstract. To design a set of fast-response and high-accuracy Hawk-eye judging systems for tennis games. Method: Quantum particle swarm optimization is used to calibrate the camera to take pictures of tennis movement, and then these pictures are matched with three-dimensional. Finally, the motion trajectory of tennis is reduced by Gaussian fitting. In the 100 sets of data, the accuracy rate of the system judged within 5 cm from the border line was about 80 %, and the accuracy rate of other regions was more than 95 %. The average time of the system was also within 10 s. The Hawk-Eye system based on quantum particle swarm optimization calibration camera has the advantages of fast response and high accuracy of judgment, which is suitable for high-level tennis game.

Key words. Hawk-eye, tennis competition, quantum particle swarm optimization.

1. Introduction

With the development of society, science and technology are also constantly moving forward. Tennis players' own physical fitness and technology as well as their own equipment are increasing, and the speed of tennis also will be accelerated, even to the speed that people is difficult to see clearly with the eyes. So, there are inevitable erroneous judgments and commutations in the tennis game. Especially in high-level events, the game decision made by the information observed by the chief umpire in accordance with the naked eye at the scene is difficult to make the game fair. Because there are limits and blind spots in human vision. Hawk-eye technology tracks the path of the tennis movement through the camera, and calculates and synthesizes a series of collected data, and then clearly shows the audience the final movement path of the tennis and landing point. It can overcome the blindness of the human eye so that the audience can watch the game from more angles, while making the referee's decision more accurate to ensure the interests of the players [1]. The accuracy of the Hawk-eye system should be as high as possible in order to make the game fairer. The response speed of the Hawk-eye system should be as

---

[1] Xi'an Physical Education University, 710068, Shaanxi, China; E-mail: `jff_jiafufeng@163.com`

fast as possible in order not to interrupt the continuity of the game. Based on this purpose, we studied and designed the tennis Hawk-eye system based on the quantum particle swarm optimization by using the virtual instrument platform and the image processing and simulation technology.

2. Theoretical review

2.1. Hawk-eye system

The Hawk-eye is an instant replay system invented by Britishman Paul Hawkins to help the audience overcome many of the blind spots in the viewing angle, enabling viewers to clearly watch the multi-perspective tennis game process through the playbacks provided by the Hawk-eye technology [2]. So, for the audience, the Hawk-eye technology has given them a good visual experience. The set of Hawk-eye technology is very precise. The whole system has 8 or 10 high-speed cameras, which divides the three-dimensional space of the entire competition venues into multiple areas that the measurement units is millimeter with the help of the computer. So, it can capture the basic data of the trajectory of tennis on the field from different angles through high-speed camera, and will observe the blind spot and dead angle that the human eye cannot see. The Hawk-eye technology system also has four computers and a large screen. The captured data will be drawn as a three-dimensional image after computer calculates, and the tennis line and placement will be presented in the big screen to the audience by using the timely imaging technology. Hawk-eye technology is very efficient, the process from the collection of data to demonstration of the results only takes less than 10 seconds. Therefore, the Hawk-eye system is now widely used in a variety of television broadcasts.

2.2. Particle swarm optimization

Particle Swarm Optimization (PSO) [3] is an evolutionary computing technique and derives from the study on the prey behavior of birds. It is a kind of optimization tool based on iteration, which is similar to genetic algorithm. The system is initialized to a set of stochastic solutions, and it interactively searches for the optimal solution. The birds in the population are abstracted as "particles" without quality and shape. We find the optimal solution in the complex solution space through the cooperation and information sharing of these "particles".

In the PSO, the solution to each optimization problem can be considered as a bird in the search space. We call it "particles." All particles have a fitness value determined by the optimized function, and each particle has a velocity that determines the direction and distance they fly, and the particles will search following the current optimal particle in the solution space.

PSO is initialized to a group of random particles, and then find the optimal solution through iteration. In each iteration, the particle updates itself by tracking the two extremes. One is the best solution that the particle finds by itself, called the individual extremum pbest; the other extreme is the optimal solution gbest of all the particles in the whole particle population, which is the global extreme value. In addition, we cannot use the entire population but only a part to be the neighbors of the particle, then the extreme value among all neighbors is the local extreme [6]. At the same time, each particle constantly changes its speed in the solution space to determine its own direction and flight distance, and "fly" towards the area that the pbest and gbest point to as far as possible.

In the mathematical model of the PSO algorithm, the bird is abstracted as a "particle" without mass and volume. The solution group is equivalent to a bird group, the "good news" is equivalent to the optimal solution of the evolution of each generation, the food source is equivalent to the global optimal solution, and the migration from one place to another is equivalent to the evolution of solution group. The evolution or evolution of the whole system includes the emergence and differentiation of the new levels and emergence of diversity, and the emergence of new themes.

2.3. Quantum particle swarm optimization

After the PSO algorithm, scholars have also carried out a lot of work on improving the convergence and diversity of the algorithm. Even though a variety of improved algorithms are proposed on the basis of the PSO algorithm, the PSO algorithm itself is flawed, and the most fundamental of these is that it has been proved not a globally convergent algorithm [5]. The authors put forward quantum particle swarm optimization algorithm on the basis of PSO algorithm in order to solve this problem. The properties of it is completely different from the particles in the quantum space. It can be searched in the whole feasible solution space, so the global search performance of the quantum particle swarm optimization is much better than that of the general PSO algorithm.

Quantum Particle Swarm Optimization (QPSO) follows the PSO. The state of the particle is no longer expressed by the position vector and the velocity vector in the quantum space, but is described by the wave function. Owing to the uncertainty principle, the position and velocity of the particles cannot be accurately determined at the same time. Particles can be examined in the spatial representation without considering the velocity. The probability that a particle appears at position x is represented by a probability density function that is not limited to a given orbital function [4].

3. Method

3.1. Calibration of the camera

The camera is the front end of the whole system, and the information quality has great influence on the subsequent image processing and 3D construction. In the calibration process of the camera, the quantum particle swarm optimization algorithm is introduced to overcome the problem that the system is easy to fall into local small and the error is large, and the venue location of the camera is shown in

Fig. 1.

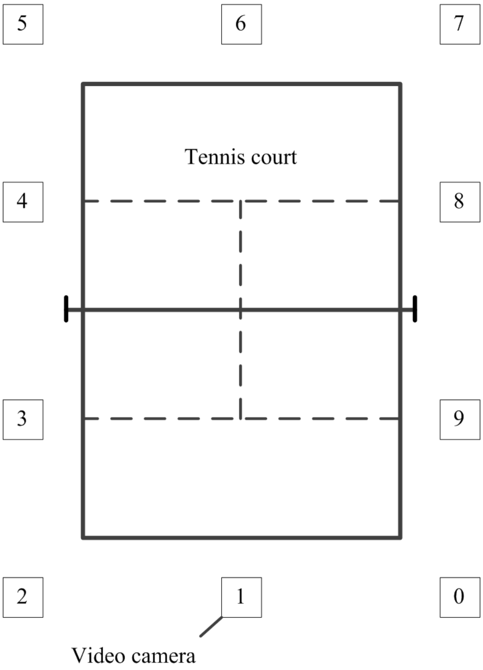

Fig. 1. Plan view of the camera arrangement

The relationship between the 3D point M and the 2D point m is:

$$s\tilde{m} = A[Rt]\tilde{M}\,, \tag{1}$$

$$s\begin{bmatrix} u \\ v \\ l \end{bmatrix} = A[r_1, r_2, r_3, t]\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = A[r_1, r_2, t]\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}\,. \tag{2}$$

Here, A is the camera parameter matrix, $\tilde{M}$ is the homogeneous matrix of the three-dimensional point, $\tilde{m}$ is the homogeneous matrix of the two-dimensional point, and r_1, r_2, r_3, t is the rotation matrix and translation of the camera relative to the world coordinate system. When $Z = 0$, then

$$H = [h_1, h_2, h_3]\lambda K[r_1, r_2, t]\,, \tag{3}$$

$$r_1 = \frac{1}{\lambda}K^{-1}h_1\,, \quad r_2 = \frac{1}{\lambda}K^{-1}h_2\,. \tag{4}$$

From the properties of the rotation matrix, the constraints of the camera parameter matrix are

$$h_1^{\mathrm{T}} K^{\mathrm{T}} K^{-1} h_2 = 0, \quad h_1^{\mathrm{T}} K^{\mathrm{T}} K^{-1} h_1 = h_2^{\mathrm{T}} K^{\mathrm{T}} K^{-1} h_2 \,. \tag{5}$$

When the number of captured images is greater than 3, K can be solved linearly to obtain the internal parameters. The quantum particle swarm optimization equation is

$$v_i(t+1) = v_i(t) + c_1 \mathrm{rand}_1 [p_i(t) - x_i(t)] + c_2 \mathrm{rand}_2 [p_g(t) - x_i(t)] \,, \tag{6}$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \,. \tag{7}$$

Here, i is the index of the particle, t is the evolution time, and rand_1, rand_2 are the random functions. Symbol c_1 is the optimal step size of the particle itself, c_2 is the global optimal step size of the particle, and v_i is the velocity of the i-th particle.

Then the evolution equation is obtained through the basics of the quantum mechanics, and the final global limit is the optimal value of the camera calibration.

$$\begin{aligned} & x_i(t+1) = p_i \pm \beta \left| m_{\mathrm{best}} - x_i(t) \right| \ln(1/\mu) \\ & p_i = \tfrac{\mathrm{rand}_1 \cdot p_i + \mathrm{rand}_2 \cdot p_g}{\mathrm{rand}_1 + \mathrm{rand}_2} \\ & m_{\mathrm{best}} = \textstyle\sum_{i=1}^{n} p_i / n = \sum_{i=1}^{n} p_{i1} / n = \ldots = \sum_{i=1}^{n} p_{i\mathrm{D}} / n \end{aligned} \tag{8}$$

3.2. Two-dimensional detection and tracking of tennis

The two-dimensional detection of tennis is achieved by the background difference method. In order to reduce the influence of noise, illumination and so on, the Gaussian template is used to extract the background. First, the component has the background model $A[x,y,t] = \left| \mu \left| \sigma^2 \right| \right|$ of the Gaussian distribution, μ is the average value of the pixels, and σ^2 is the variance. The pixel in current frame of camera is set as $I[x,y,t]$, and transform according to the following equation

$$\mathrm{Acc}[x,y,t] = (1-a)\mathrm{Acc}[x,y,t-1] + aI[x,y,t] \,, \tag{9}$$

The constant a is 0.5, after the accumulation, the value of a pixel in current frame is larger than the past weight.The MATLAB image processing tools in the subtraction is used, and the image is binarized by rgb2gray function. Then the subtraction operation is carried out through the current frame and background image, and then is threshold segmentation, you can extract the target parameters, such as coordinates, length and width [7].

In the dynamic tracking of tennis, the CAMshift method is used, and the video image is transformed into color probability distribution according to the YUV information. The tennis characteristic parameters obtained by the difference method are taken as the initial parameters of the search, and set as $I[x,y]$, to calculate the zero order matrix, the first order matrix M_{01}, the second order matrix M_{02}, M_{20}

and M_{11}, and then calculate the current centroid position as

$$(x_{\mathrm{c}}, y_{\mathrm{c}}) = \left[\frac{M_{10}}{M_{00}}, \frac{M_{01}}{M_{00}}\right]. \tag{10}$$

The result is taken as the center of the search window in the next frame, and then the size of the search window is adjusted according to M_{00}:

$$\begin{aligned} l &= \sqrt{\tfrac{(a+b)+\sqrt{b^2+(a-c)^2}}{2}}, \\ w &= \sqrt{\tfrac{(a+b)-\sqrt{b^2+(a-c)^2}}{2}}, \end{aligned} \tag{11}$$

where l is the length of the search window, w is its width, $a = M_{20}/M_{00} - x_{\mathrm{c}}^2$, $b = 2[M_{11}/M_{00} - x_{\mathrm{c}}y_{\mathrm{c}}]$, $c = M_{02}/M_{00} - y_{\mathrm{c}}^2$. Calculating according to this cycle until the distance between center of mass and the core is less than the threshold, one can get the moving target parameters.

3.3. Reconstruction of three-dimensional points

It needs at least two different angles of the image in order to achieve the three-dimensional world coordinates of the positioning sphere. The 3D point is reconstructed in the ideal state from the projection point of the pixel coordinates, as shown in Fig. 2.

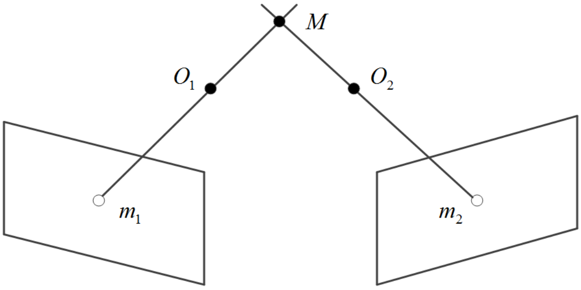

Fig. 2. Ideal 3D point reconstruction

The optical center O_1 through the camera can draw a ray to the space from the projection points m_1 and m_2 of the first image point of space. The optical center O_2 through the camera can draw another ray to the space from the projection of the second image point of space. The intersection point of the different rays drawn from different dimensions is the space point in the real world [8]. However, as shown in Fig. 3, because the images of different angles have image noise and camera calibration error, so that the intersection point on the space does not exist, and the three-dimensional reconstruction of the point problem turns into the problem about selecting the optimal solution in the point space. There are many mature algorithms for this optimal solution problems, such as the least squares method, it can be achieved by using iterations.

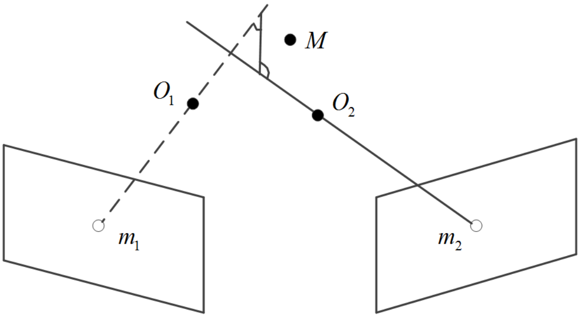

Fig. 3. Real 3D point reconstruction

After the coordinates of the spatial points are obtained, these discrete points need to be fitted. The Gaussian fitting is used to fit the corresponding points of the image using Matlab ImageProcessing. The larger area is the intersection of the centerline and the edge. The system chooses 10 pairs of points and executes TFORM = cp2tform (input points, base-points, transfromtype), B = imtransform (A, TFORM), A, B are different camera images. The curve fitting principle is the least squares method, and the curve with the quadratic sum of the distance of the given point is the smallest, and one can get the trajectory by using Matlab toolbox Gaussian.

4. Results

We create the three-dimensional space with three-dimensional coordinates by using Grid Properties.vi, and take XY plane as the ground, and create ball of the tennis through Create Sphere.vi and Create object.vi. The ball is running in accordance with the fitting track. We selected 50 points to simulate the ball running trajectory in order to make the display more coherent, and created another scene display to show the top view. In the top view, the audience can clearly see the tennis landing point, and the coordinates of the landing point and the tennis ball radius are same. When it beyond the border, it is judged out of bounds, determining the bottom line $Y = 0$. If the coordinates of the tennis landing point on the X axis, then it is positive, and the ball is in bounds. If it is negative, then it is judged the ball is outside the bounds.

100 sets of measurement data were selected for testing in order to verify the correctness of the system. The results show that the average response time of the system is 9.12 s, and some results are shown in Table 1.

The relationship between the correct rate judged by system and the distance from the boundary is shown in Fig. 4.

From the above results, the correct rate judged by the system was 80 % when it is -5∼5 cm from the border, and other places have reached more than 95 %, and the average processing time of 10 s. The decision and the processing time meet the design requirements.

Table 1. Test results

Groups	Actual distance (cm)	Actual judgments	Distance obtained by system (cm)	Judged by system
1	9.3	No	9.5	No
2	10.2	No	10.1	No
3	–7.1	Yes	–7.4	Yes
4	4.8	No	5.1	No
5	12.4	No	12.7	No
6	–15.1	Yes	–14.8	Yes
7	–0.4	Yes	0.2	No
8	5.9	No	6.2	No
9	13.8	No	13.5	No
10	3.2	No	3.5	No
⋮	⋮	⋮	⋮	⋮
95	10.2	No	9.9	No
96	7.9	No	7.8	No
97	–8.5	Yes	–8.3	Yes
98	–6.3	Yes	–6.4	Yes
99	4.9	No	4.7	No
100	–8.2	Yes	–8.3	Yes

5. Conclusion

The camera calibration based on p quantum particle swarm optimization can overcome the shortcoming that the system is easy to fall into the local small and the back-projection error is large, and the calibration precision of the camera is improved. Based on this algorithm, the tennis Hawk-eye system can track all the trajectories of tennis movement and the accurate track of the landing point of the tennis ball. It can determine the result, its level of visualization is high, the speed of judgment is fast and the result is accurate, so it can meet the design requirement.

References

[1] S. Mecheri, F. Rioult, B. Mantel, F. Kauffmann, N. Benguigui: *The serve im-*

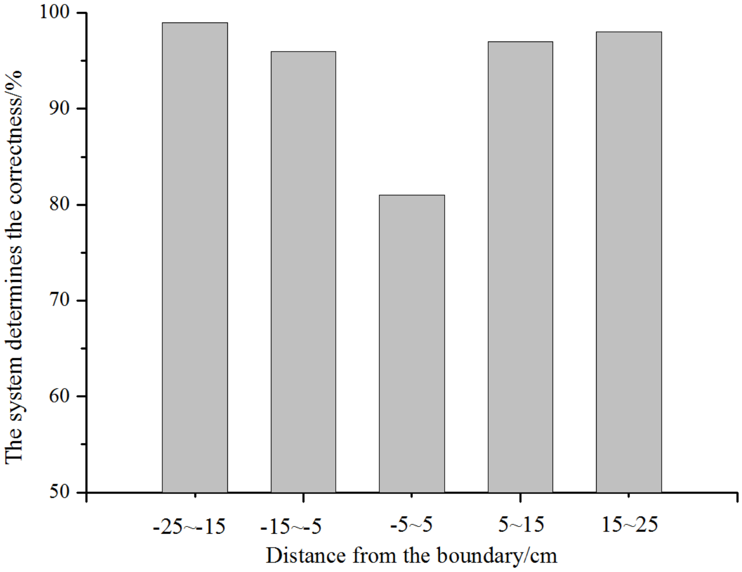

Fig. 4. Relationship between the correct rate judged by system and the distance from the boundary

pact in tennis: First large-scale study of big Hawk-Eye data. Journal Statistical Analysis and Data Mining *9* (2016), No. 5, 310–325.

[2] J. Cui, Z. Liu, L. Xu: *Modelling and simulation for table tennis referee regulation based on finite state machine.* Journal of Sports Sciences (2016), 1–9, paper 1241417.

[3] J. Kennedy: *Particle swarm optimization.* Encyclopedia of machine learning, Springer US (2011), 760–766.

[4] Y. F. Ji, C. L. Wang, Z. H. Shi, J. Ren, L. Zhu: *Table tennis service umpiring system based on video identification—for height and angle of the throwing ball.* Wearable Sensors and Robots, Springer, Singapore, Lecture Notes in Electrical Engineering *399* (2017), 495–509.

[5] Y. Hao, Z. Hao, H. Yang: *Table tennis placement is determined based on artificial intelligence.* Journal of Residuals Science & Technology *13* (2016), No. 5, 16.1–16.3.

[6] D. Wang: *Application of simulation technology in table tennis flight theory model.* Journal of Computational and Theoretical Nanoscience *13* (2016), No. 12, 9793–9797.

[7] C. Brechbuhl, G. Millet, L. Schmitt: *Accuracy and reliability of a new tennis ball machine.* Journal of Sports Science & Medicine *15* (2016), No. 2, 263–267.

[8] P. Karuna, M. Preeti: *Global plagiarism management through intelligence of hawk eye.* Indian Journal of Science and Technology *9* (2016), No. 15, 1–10.

Received May 7, 2017