Low-cost Tennis Line Call System with Four Webcams

Yu-Po (Ken) Wong
Department of Applied Physics, Stanford University
Stanford, CA 94305
mkenwong@stanford.edu

Abstract

In this project, a low-cost tennis line call system with ball speed detector is implemented and tested. I demonstrated trajectory calculation for low-speed ball and high-speed ball (70mph) without player. Ball detection with low-speed ball with player is also demonstrated. The precision of the system at T-point is measured at 8mm. This project explore the possibility of an affordable and portable tennis line call system with speed detection.

1. Introduction

Hawk-eye system, developed by Dr. Paul Hawkins in 2001, is a trajectory-tracking system used in numerous sports. It used videos from six high performance cameras to reconstruct 3-dimensional representation of the trajectory of the ball. [1] It is used in challenge system in professional tennis matches since 2006 to verify close line calls which might be difficult for human line judges. The system costs about 60k to 70k USD, and required the cameras to be mounted at two to three stories height. A typical visualization result from hawk-eye system is shown in Figure 1. It is neither an affordable nor portable system.

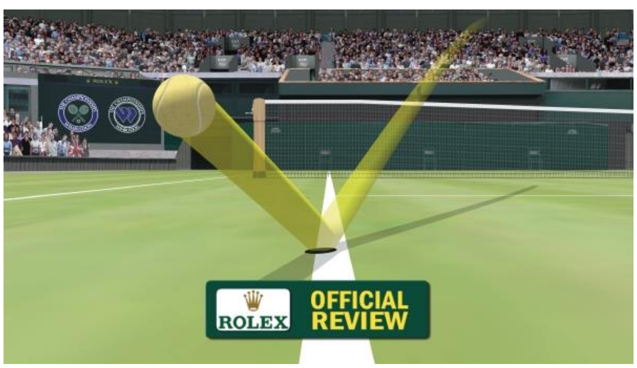

Figure 1: A typical visualization result from Hawk-eye system

For tennis ball speed measurement, tennis ball radar gun is usually used to only measure serve speed in professional tennis match. They don't track the ball speed on different aspect of the game, like swings and smashes.

For recreational tennis matches, players usually play without line judges and never with a hawk-eye system. It is difficult for active players in the match to determine most of the line calls. Furthermore, recreational players usually don't have any idea at the exact speed of their serve, swing, and smash speed.

In this project, I'd like to explore the possibility of an affordable and portable system using four webcams to do line call in tennis games as well as calculating the speed of the ball on all the aspects of a tennis game.

2. Previous work

2.1. Review of previous work

Professional Hawk-eye system costs 60 to 70k USD, using 10 really high speed camera at 340fps mounting no second floor. The precision of the system is up to 2.6mm. The details about their method and algorithms are not open to public [1].

S. V. Mora et al. implemented a system with four high speed (60fps) and high resolution (1280x1024 pixels) cameras at the four corners of a tennis court with external 60Hz synchronization signal. The cameras are mounted at four corners of a tennis court by tripods. They used background subtraction, different between consecutive frame, and the ball trajectory information from earlier frames to conduct ball detection. For precision analysis, they used reprojection error after fitting to get about 20 pixels for 80 percent of the frames. The estimated cost of the system is 4k USD.

2.2. Comparison between this project and previous work

In this project, I use 4 Logitech C615 webcam at 800x600 resolution and 30fps. The system only cost $224 USD. My cameras are mounted on the tennis court posts instead of mounted on second floor or on tripod. Instead of high performance and fancy equipment, this project aims on affordability and portability.

I implement background subtraction, frame difference, morphological image processing (erosion and dilation), and big object removal to track the ball. The methods and algorithm are not as good as reference 2. In this project, I focus on building up the system to evaluate the possibility of a low-cost recreational player friendly tennis line call and tennis ball speed measurement system.

3. Technical Solution

3.1. Summary of technical solution

Four main parts are implemented in this project: camera calibration, ball detection, triangulation, and curve fitting.

3.2. Technical details

3.2.1 Hardware setup

In this project, we only work on one side of the tennis court. 2 Logitech C615 webcam at 800x600 resolution and 30fps is connected to a PC with 10m long active USB extension cables. The cameras are mounted on the tennis court post (Fig. 2). An example of stereo images from the system is shown in Figure 3.

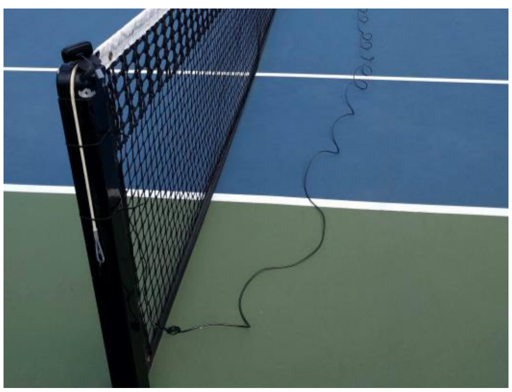

Figure 2: Hardware setup. A Logitech C615 mounted on tennis post

3.2.2 Camera Calibration

Camera internal matrix is determined by using MATLAB Camera Calibrator with calibration rig. The internal matrix for two cameras are:

$$K_L = \begin{bmatrix} 806.9 & 0 & 413.6 \\ 0 & 806.8 & 310.3 \\ 0 & 0 & 1 \end{bmatrix}$$
$$K_R = \begin{bmatrix} 835.6 & 0 & 397.0 \\ 0 & 835.3 & 303.4 \\ 0 & 0 & 1 \end{bmatrix}$$

For the external matrix, I used some intersections of court lines and some measured points on the back wall to do the calibration (Fig. n). The 3D position in world coordinate and 2D position in image coordinate are manually measured. An algorithm is used to figure out both translational (t) and rotational part (R):

1. For a given t, the rotational part can be calculated by solving the linear system $R'P_i = P_i'$, where $P_i' = K^{-1}p_i/\|K^{-1}p_i\| \times \|P_i\|$. Then $R = UIV^T$, where USV^T is the SVD of R' and I is the identity matrix
2. Set up a function as $\varepsilon(t)$ is the reprojection error for the calculation in step 1

With this algorithm, the reprojection error of the camera model is 3.776 pixels.

Figure 3: Example of stereo images from the system

3.2.3 Ball detection

i. Without player

Ball detection without player can be done by simply doing background subtraction with a Gaussian fitting.

ii. With player

Ball detection with play is much more complicated. Here are the summary of the algorithm I used to locate the ball:

1. Background subtraction with a threshold gives us a map of pixels (M_b) with objects on it

2. Do dilation and erosion on M_b to connect the pixels around the player.

3. Do big object removal to remove player from M_b to get $M2_b$

4. Difference between consecutive frames with threshold gives us a map (M_d) of pixels with moving objects on it

5. Calculate a refined map M_t = and($M2_b$, M_d)

6. Since the ball is moving, pixels around M_t from previous frame cannot be the position where the ball

is. Build a map M_{pre} to clean up those pixels. This will clean up the pixels around the player which is not detected by big object removal

7. The final map will be M_f = and(M_t, M_{pre})

This method did a decent job to locate the ball when the ball is not too close or overlap on the player. All the maps for a frame is shown in Figure 4.

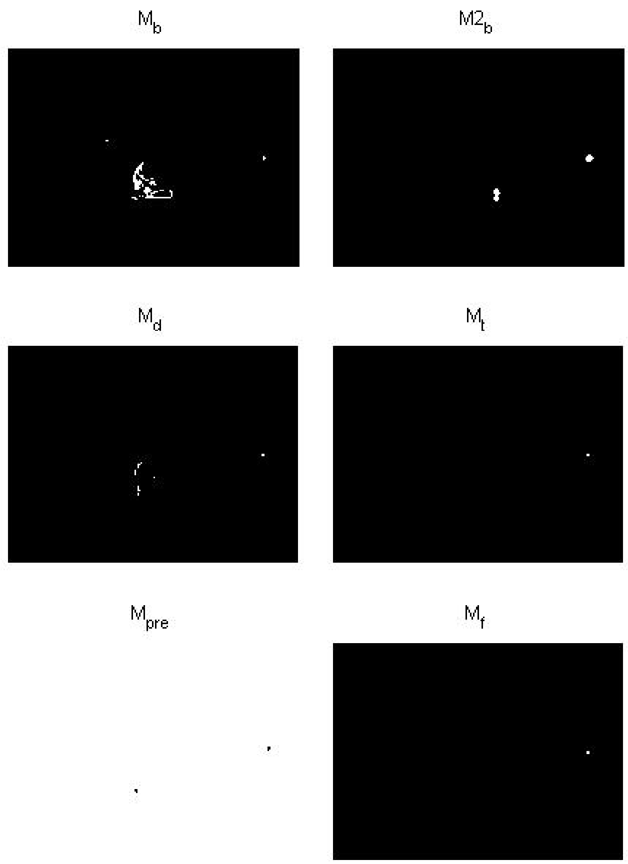

Figure 4: Maps used in ball detection with player on court.

3.2.4 Triangulation

Triangulation is done by minimizing $\|p_L - M_L P\| + \|p_R - M_R P\|$ with respect to P. The average triangulation error is 1.0014 pixels.

Since the low-cost webcam doesn't have external trigger, the frames are not synchronized among cameras. An interpolation method is proposed in the project progress report to use the timestamp to align the frames from different cameras. I tested the method, but it's working well at all due to the discontinuity of the curve at the impact point, which is the point where the ball contact the ground. An example of the interpolation is shown in Figure 5.

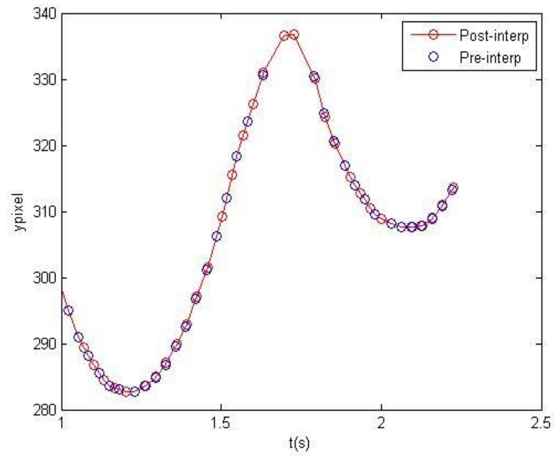

Figure 5: Result from interpolation between two camera timestamps to correct unsynchronized nature for low-cost webcams. The trajectory is supposed to be sharp at turning point (intersection of two parabolas). Interpolation makes the trajectory smooth. Special designed interpolation is required here.

3.2.5 Information extraction

After triangulation, we have (x,y,z,t) information for each frame.

i. Impact point information

I split the (x,y,z,t) information into two parts: (x,y,t) and (z,t). For (z,t) part, I fit parabola to different sections of the curve to figure out the time of ball contacting the ground. Fitting is also done on (x,y,t). With both fitting information. The position of impact point (x',y') can be calculated.

ii. Speed information

Speed information can be extracted by calculating difference of (x,y,z,t) between two frames.

4. Experiments

4.1. Low speed ball without player

The original (x,y) and (z,t) plot is shown in Figure 6. Fittings are done on both of the plot and shown in Figure 7. For (z,t) fitting, the position of the intersection of the curves are 42mm and 57mm, where the actually radius of a tennis ball is 32.7~34.3mm. For (x,y) fitting, the rms fitting error is 3.24mm. The speed of the ball is measured at 19mph.

4.2. High speed ball without player

The original (x,y) and (z,t) plot is shown in Fig. 8. The speed is shown in Fig. 9. The speed of the ball is up to

70mph.

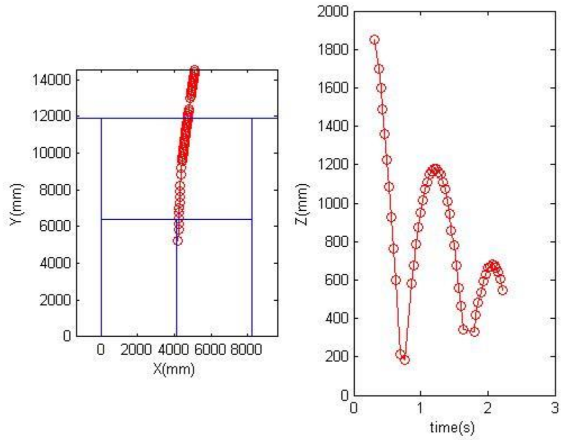

Figure 6: (x,y,z,t) information for low speed without player test. Left: (x,y) curve. Right: (z,t) curve.

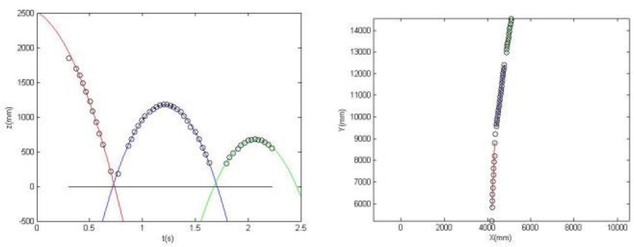

Figure 7: (x,y,z,t) fitting information for low speed without player test. Left: (z,t) curve fitting. Right: (x,y) curve fitting for three according to three parabola in (z,t) plot. Average rms fitting error is 3.24mm.

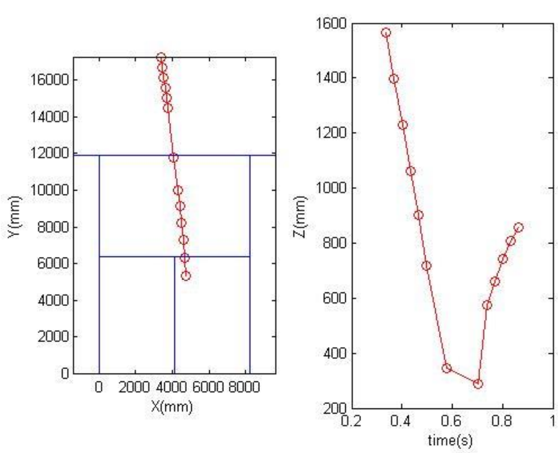

Figure 8: (x,y,z,t) information for high speed without player test. Left: (x,y) curve. Right: (z,t) curve.

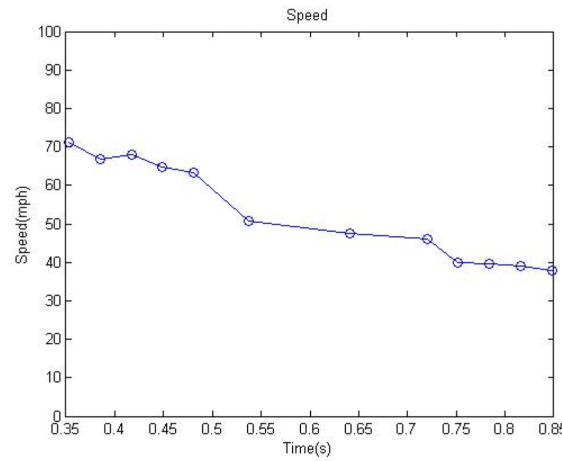

Figure 8: speed information for high speed without player test. The ball speed after smash is about 70mph.

4.3. Low speed ball with player

The original (x,y) and (z,t) plot is shown in Fig. 10. In this case, we lost ball position in many frames since it's pretty easy to get ball overlap on the player in one of the camera and triangulation needs ball information from both cameras. Figure 11 shows some frames that only one of the camera or neither can detect the ball.

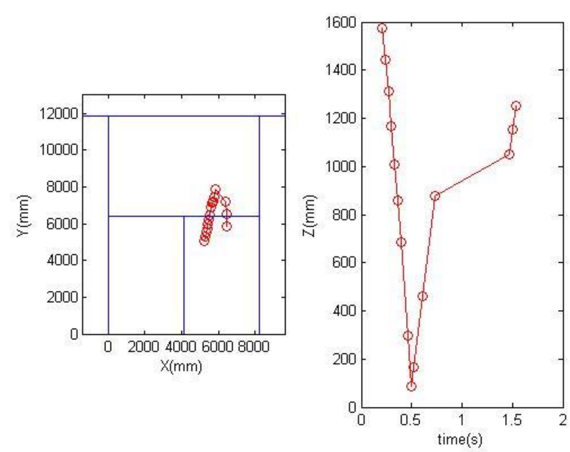

Figure 10: (x,y,z,t) information for low speed with player test. Left: (x,y) curve. Right: (z,t) curve.

Figure 11: Two frames that only one of the camera or neither can detect the ball

4.4. Precision and Accuracy Test

A test is proposed to test the precision and accuracy of the system. A ball is dropped onto the T-point, which is the intersection of service line and center line, of the court by releasing by hand without initial velocity in any direction. The ball should just fall as a free falling object and have the same (x,y) coordinate as the T-point.

The (x,y,z,t) information is shown in Figure 12. The zoom in information with fitting is shown in Figure 13.

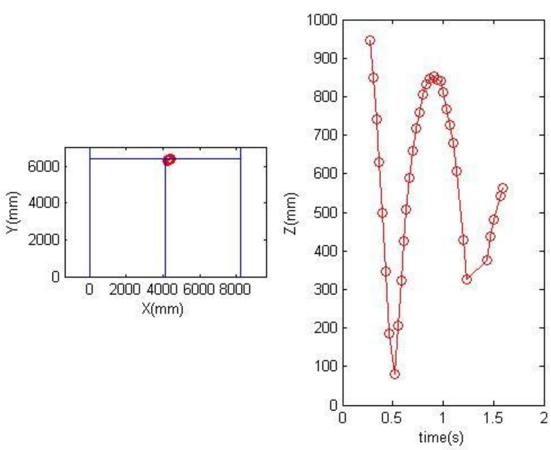

Figure 12: (x,y,z,t) information for precision and accuracy test. Left: (x,y) curve. Right: (z,t) curve.

We noticed that it's really difficult to release a ball with absolute zero initial velocity. Aiming for the T-point is even more difficult. For this test, I minimized the initial velocity, but the contact point is not on the T-spot when I did the experiment. Furthermore, due to a tennis ball is not perfectly circular (there's the white silicon line on the ball), the ball tend to gain some more lateral velocity after contacting the ground.

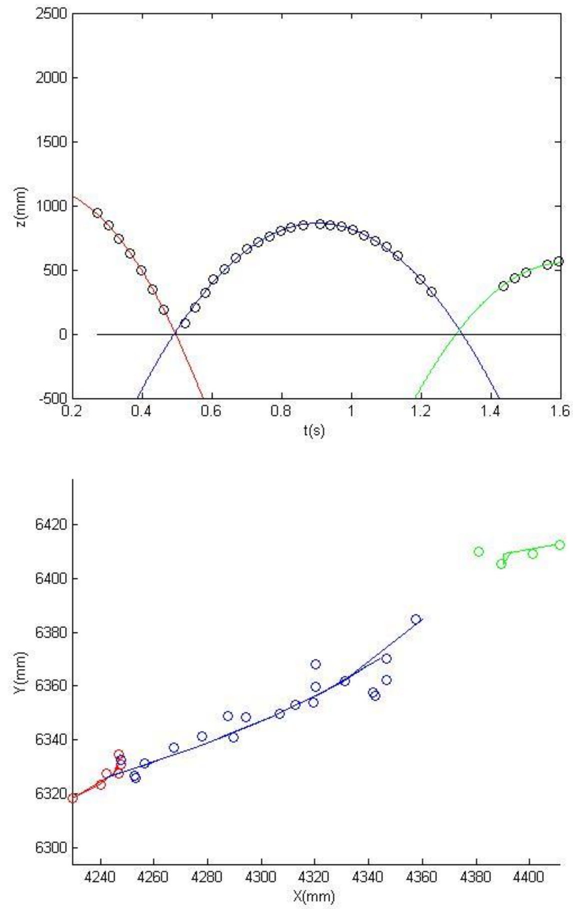

Figure 13: Zoom-in and fitting for the (x,y,z,t) information for precision and accuracy test. Top: (z,t) curve. Bottom: (x,y) curve.

For the accuracy test, we don't have a valid result here due to the ball was not dropped onto the T-point. A more careful designed experiment needs to be carried out to test it. Due to the issue with gaining more lateral velocity at the contact point, we calculate the statistics of (x,y) distribution for the first 7 (red region) and first 29 (red+blue region) frames. The standard deviations of distance from center is: 8mm (first 7 frames) and 46mm (first 29 frames). The 8mm is closer to the actual precision of the system since we can clearly see in Figure 13 that the ball gain a lot more lateral velocity after contacting with the ground.

4.5. Code

https://www.dropbox.com/sh/eganwiptrudddlz/AAAZyfQkbXkW4AR5WisxeSoIa?dl=0

camera calibration: run_calibCam_fun.m
ball detection without player: saveLocateBall.m

ball detection with player: saveManextract.m, saveLocateBall_man.m
triangulation and analysis: calBall_ken.m

5. Conclusions

I demonstrate a low-cost system with four Logitech C615 webcams with a total cost around $200 USD. The system has 8mm precision so far. The accuracy of the system needs to be further studied. The ball trajectory calculation is demonstrated with low speed and high speed ball. The result is quite promising. The speed of the ball can be measured. And we measured up to 70mph. Ball detection with player on court is demonstrated but has limited performance.

The performance of the system can be further improved by fixing the unsynchronized problem among cameras by interpolation. A better ball detection technique with player on court needs to be developed. Automated camera calibration is another important feature to be explore.

6. References

[1] http://www.hawkeyeinnovations.co.uk/sports/tennis

[2] S. Vinyes Mora, G. Barnett, C.O. da Costa-Luis, J. Garcia Maegli, M. Ingram, J. Neale and W.J. Knottenbelt. A Low-cost Spatiotemporal Data Collection System for Tennis. Proc. 5th Mathematics in Sport Conference. Loughborough, UK, June 2015.