Tracking Ping Pong Ball in Video by Matlab

Cheng-Chieh Cheng[1] (鄭丞傑), Ching-Lin Huang[1] (黃敬麟),
Min-Chi Wang[1] (王敏齊)

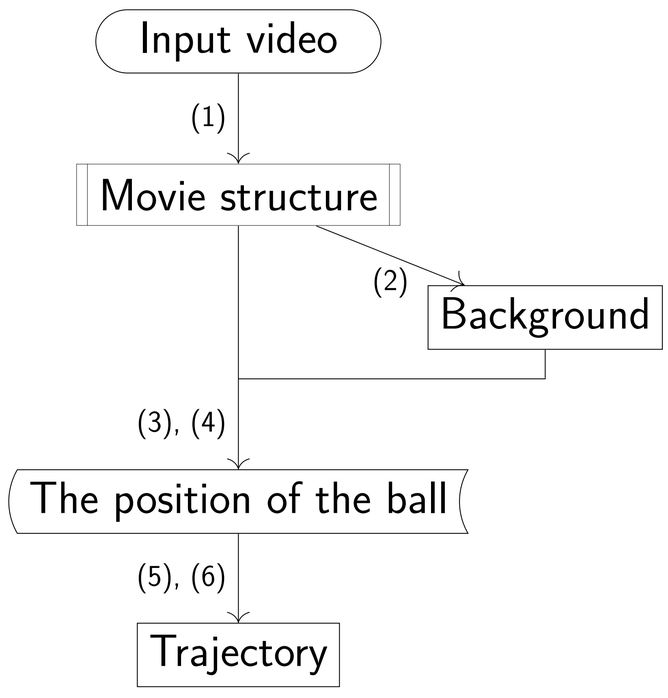
Motivation

There are Hawk-Eye system used in many sports, such as badminton, tennis and cricket. However, none is used in table tennis. Our goal is to build to analyze table tennis matches such system through which we hope people improve their table tennis skills.

Introduction

This code tracks the trajectory of the ping pong ball in video, similar to the Hawk-Eye system. It cuts video into many frames, locates the accurate position of the ball in each frame, and calculates the trajectory of the ball, where the second part is the most difficult. So most of our work focuses on detecting ping pong balls in a frame, and we hope it can also be applied on other ball-games.

Main Structure



(1) getStructure(video): Construct a structure, called movie, which saves the data of video and position of balls.

(2) getBackground(movie): To get the background.

(3) detectBalls(frame, background): To get the binary image of balls.

(4) getPosition(ball_matrix): To get the position of balls.

(5) removeOthers(movie): To remove the detected object which is not a ball.

(6) makevideo(movie): To make the video of the moving balls.

Method

Make a good background image to compare with other frames.

1. Take difference of two successive frames.
2. Use morphological methods to remove small component.
3. Replace those pixels in difference by another pair of frames without difference in those pixels.

Find the position of the ball.

1. Find the difference of the frame and background.
2. Calculate the variance of the difference of all pixels.
3. Get a binary image determined by variance.
4. Modify the binary image by some morphological methods. (See the beginning of next section.)
5. Find the ball-like component with the highest variance.
6. Label it as a ball.

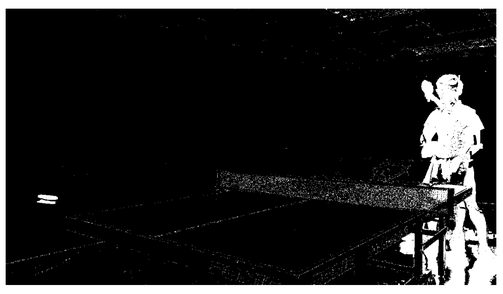
Since we always get a ball-like component even though there is no ball. Thus, we delete the label if the difference of the positions of two balls in successive frames are too big.

Algorithm

Image modification in detectBalls():

Original image.

Get the binary image determined by the variance.



Use bwareaopen() to remove small objects.

Use imclose() to fill the gap.

Use bwareaopen() to remove big objects.

Use bwlabeln() to get the ball.

We use the following functions of morphology to detect the ball.

- Dilation $A \oplus B := \bigcup_{b \in B} A_b$, where $A_b = \{a + b \colon a \in A\}$.
- Erosion $A \ominus B := \{z \colon B_z \subset A\}$.
- Closing $A \bullet B := (A \oplus B) \ominus B$.
- imdilate(A,B): a function in matlab dilating A by B.
- imclose(A,B): a function in matlab closing A by B.

Attempts

There are only few materials about this subject, so we struggled when started.

- We tried to subtract the successive two frames to delete the background image. But after subtracting, not only the ball but also the server were remained. So the computer was still unable to find the ball.
- We applied various filters to highlight the ball, which unfortunately, also highlighted other bright part of frames. So we still failed.
- We captured the ball in a frame as the sample. And for every position in other frames, we calculated the sum square of difference between the sample and its neighborhood, and found the position that minimized the sum square. However, it took 10 mins for each frame, which was of course unacceptable.

Prospects

- Adding some functions to improve the accuracy of estimated position of ping pong balls.
- Adapt our code to detect the position of balls in playing since we now can only analyze serves.
- Collecting large training data of trajectories of different categories of balls. Apply deep learning to produce a code which can categorize each play.
- Quantify the difference of our plays and the athlete's to be an index of the people's table tennis skills.

Reference

1. https://github.com/Trujillo94/Tracking-a-ball-with-matlab
2. https://en.wikipedia.org/wiki/Otsu%27s_method
3. http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/OWENS/LECT3/node3.html

Our code

https://github.com/Katasuke/ICM_G14

[1]Department of Mathematics, National Taiwan University. Email: b05201051@ntu.edu.tw, b05201010@ntu.edu.tw, b05201025@ntu.edu.tw, respectively.