The Ping-Pong Project

Adhithya Ravishankar

University of Mississippi. Sally McDonnell Barksdale Honors College

Follow this and additional works at: https://egrove.olemiss.edu/hon_thesis

Part of the Computer Sciences Commons

Recommended Citation

Ravishankar, Adhithya, "The Ping-Pong Project" (2017). Honors Theses. 11.
https://egrove.olemiss.edu/hon_thesis/11

This Undergraduate Thesis is brought to you for free and open access by the Honors College (Sally McDonnell Barksdale Honors College) at eGrove. It has been accepted for inclusion in Honors Theses by an authorized administrator of eGrove. For more information, please contact egrove@olemiss.edu.
The Ping Pong Project
by
Adhithya Ravishankar

A thesis submitted to the faculty of The University of Mississippi in partial fulfillment of the requirements of the Sally McDonnell Barksdale Honors College.

Oxford, Miss.
May 2017

Approved by

[Signatures]

Advisor: Professor Yixin Chen
Reader: Professor Naeemul Hassan
Reader: Professor Dawn Wilkins
Dedication

To my dad, who I will always remember as my biggest supporter
Acknowledgements

First, I would like to thank the Sally McDonnell Barksdale Honors College for giving me the opportunity to write a thesis. I know that I would not have written one if it was not required by them, and I somehow know that it will become the crowning achievement of my undergraduate education. The Honors College has given me access to a wide range of opinions, feelings, people, places, ideas, facts, philosophies, and topics. I would also like to thank them for giving me the opportunity and resources to travel to Boston to learn from Dr. Ethan Danahy of Tufts University and Mr. Carl Vondrick of Massachusetts Institute of Technology. Not only did I learn about this research project and the future of computer science, I learned about how this topic is slowly changing conversations across the world.

Second, I would like to thank the Computer Science department for putting up with me and my antics for these past five years. My first class at Ole Miss was CSci 251, a class I would never need as I later learned, but it is a class that made me realize that computer science was my major and what I wanted to do for the rest of my life. I would like to thank Dr. Yixin Chen, Dr. Dawn Wilkins, Mr. Joseph Carlisle, Dr. Byunghyun Jang, and Dr. J. Adam Jones. Most of all, I would to thank Dr. Kristin Davidson, whose classes made me realize my desired major and whose various pieces of advice have stuck with me throughout the years, through the dark times and the fun times.

Third, I would like to thank my family. I would like to thank my mother and brother, who I have probably driven insane these past four years. I would also like to thank my dad, who passed away my freshman summer. I hope that I have made you proud these past three years, and my hope is that I will continue to do so throughout my life.

Finally, I would like to thank all my friends who have stuck with me through the thick and thin through all these years, and I especially would like to thank Baxter Elliott, Clark Tyner, Rosalie Doerksen, Michael Saccone, Will O’Keefe, Yujing Zhang, and Toler Presley. Without my friends, I would not be the person I am today; therefore, thank you.

Also, I am grateful for the Georgia Institute of Technology for accepting me.
List of Symbols & Abbreviations

720p 1280x720
1080p 1920x1080
2160p 3840x2160
4K 3840x2160
AVC Advanced Video Coding
AI artificial intelligence
B bytes
CPU central processing unit
CS computer science
CUDA Compute Unified Device Architecture
CV computer vision
DSLR Digital Single-Lens Reflex
fps frames per second
GB gigabytes
GPGPU General Purpose computing on GPU
GPU graphics processing unit
KB kilobytes
ISO light sensitivity
ITTF International Table Tennis Federation
HEVC High-Efficiency Video Coding
HCI human-computer interaction
HD high definition
HSB hue, saturation, brightness
HSL hue, saturation, luminosity (lightness)
HSV hue, saturation, value
MB megabytes
ML machine learning
MP megapixels
MW  megawatts
OpenCL  Open Computing Language
OpenCV  Open Source Computer Vision
p  progressive-scan
px  pixels
RGB  red, green, blue
rpm  rotations per minute
UHD  ultra high definition
W  watts
Abstract

Table Tennis, more commonly referred to as ping pong, has been a sport of the elite for over two centuries since its inception in the late 1800s. However, coaching for this sport is not usually taught in most places, and many do not know how to play this sport. To democratize the sport, and make it more accessible to everyone, this program was designed to analyze a player’s game and to give suggestions and recommendations to improve it. Using hard coded rules and tests, this program is designed to coach someone by detecting and measuring the speed and the spin of the ball. The results show that while the software might be ready for the real-world, the perfect environment and a high-end camera are necessary.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>COPYRIGHT</td>
<td>II</td>
</tr>
<tr>
<td>DEDICATION</td>
<td>III</td>
</tr>
<tr>
<td>ACKNOWLEDGEMENTS</td>
<td>IV</td>
</tr>
<tr>
<td>LIST OF SYMBOLS &amp; ABBREVIATIONS</td>
<td>V</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>VII</td>
</tr>
<tr>
<td>1 INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>2 IMPLEMENTATION</td>
<td>11</td>
</tr>
<tr>
<td>3 RESULTS</td>
<td>21</td>
</tr>
<tr>
<td>4 CONCLUSION</td>
<td>23</td>
</tr>
<tr>
<td>5 REFERENCES</td>
<td>33</td>
</tr>
</tbody>
</table>
Introduction

Learning a new sport requires many things from a person: reading the rulebook, purchasing the necessary equipment, and possibly trying the game out with another person. Many times, an individual who is observing them or playing with them acts as their coach; they usually have many times more experience than them and can give them the guidance needed to improve the sport. However, the player might not have another person to play the game with, much less one that can act in a coaching capacity. In this situation, software has turned into a democratization force (Friedman, 2012); people who have not had access to resources before now have access to software that can perform the job for them ("Democratization of Technology," 2013). Moreover, as computers have become more powerful each year, the capabilities of software have similarly increased. As artificial intelligence has grown in popularity, maturity, and performance, the uses cases have also increased ("Artificial Intelligence," 2017).

Basics of Ping Pong

History

Ping pong, more commonly known as table tennis, has its origins of British Victorian-era upper-class parlor games in the 1860s and 1870s ("Table Tennis," 2017). Many of them were make-shift games after dinner to cope with the long cold winters, especially when playing tennis outdoors would have been dreary. (Baksh, "History of Table Tennis," 2004, par. 1). The British military made versions of this game while residing in India. Over the years, many changes have been introduced that have turned this makeshift game into a traditional Olympic sport ("Table Tennis at the Summer
The balls, tables, rackets, and the International Table Tennis Federation (ITTF) have standardized the rules. ("Table Tennis," 2017).

**Rules**

As the case with any official Olympic sport, many rules help simplify the game play and systems required to understand the games. There are many rules regarding the game play and the point system ("Table Tennis," 2017).

**Materials**

First, the ball has a diameter of 1.57 inches and weight of 0.095 oz, in either a white or orange color. The most important dimensions for the table are the length and width, 9 feet by 5 feet. The paddle can have two different sides that can accomplish different things: one can be used to add speed or spin and the other can be used to reduce speed or spin. Alternatively, both sides can be used to serve one purpose. ("Table Tennis," 2017, sec. Equipment).

**How to Play**

A simpler way of thinking of how to play table tennis, without all the formal rules, is that a person serves the ball ("How to Play Ping Pong (Table Tennis)," 2017). The serve starts the round, and should hit both the hitter’s side and the receiver’s side of the court. If the ball never hits either one of these and falls out of bounds or hits the net, the point is made for the receiver. If it does hit both, the receiver now must hit the ball and send it back. However, when sending it back, the receiver only must hit the side of the court of the former sender and now receiver before going out of bounds. Therefore, it travels through this back and forth action until one person hits the net or manages to
make the other player not hit the ball back ("Handbook," 1946). Obtaining this goal can be very easy or very hard depending on both players’ skill levels.

**Project**

The project’s focus is on the ball and how it travels throughout the playing surface. The playing surface can be any color, but the playing surface we are using is green with white stripes. The orange ball color should be the one used to obtain the most contrast out of the video between the playing surface and the ball itself. To detect the spin, the logo imprinted on the ball is used and the time between two occurrences is used to obtain the spin.

**Artificial Intelligence**

Artificial Intelligence (AI) is defined as “intelligence exhibited by machines” (Russell & Norvig, 2003). This definition prompts the questions, what is intelligence? Is intelligence knowledge and the ability to acquire it? Is it what we refer to as “common sense”? Is it a personality trait, such as cunning or witty? Is it the ability to learn, or have the ability to not repeat past mistakes? Arguably, some of the most famous representations of artificial intelligence are in movies such as Iron Man, in which Jarvis helps Tony Stark fix an engine by analyzing it and determining its flaws (Favreau, director, *Iron Man*, 2008). Similarly, we are hoping to analyze a player’s gameplay and determine its flaws. There are two types of artificial intelligence: machine learning and explicit programming ("Artificial Intelligence," 2017).

**Machine Learning**

Machine Learning is something that has been attempted and used to great effect in
many other games before: chess, go ("AlphaGo," n.d.), and now recently poker (Metz, 2017). Machine Learning mimics one part of the artificial intelligence spectrum: the ability of machines to look at previous gameplay, the results, and be able to devise strategies from those past experiences. However, machine learning has limitations for many reasons, reasons which are like actual learning done by human beings. For one, machine learning requires hundreds if not thousands of iterations of annotated data samples before the machine can start making reasonable predictions. It can also be hard to get specifics from machine learning about what’s going wrong. Many machine learning models are a black box; things go in, but very rarely do we see results come out.

**Hardcoded**

AI not only relies on machine learning but can also be the result of hard-coded software that is programmed explicitly by a human, as is the case with this. Some people can ask if this is not like learning done by the brain, is this artificial intelligence? However, one can compare this to an actual animal that has instinctual responses, such as a baby turtle which knows upon hatching out of the egg, that it must make the long and dreary walk to the ocean (Netborn, 2017). Joeys (baby kangaroos) know that they must climb from being born into their mother’s pouch ("About Macropod Reproduction (Kangaroo and Wallaby)," 2015). Many behaviors in animals are not learned behaviors but are instinctive; hard-coded knowledge into the brain that one must do this when this happens.

The difference then between a regular program and a software program that exhibits artificial intelligence is the human element; the program must be doing
something that has traditionally been known in the realm of human intelligence and not something considered menial work. For example, driving a car is very hard coded with a set of rules and regulations about what happens when a change occurs on the road: a change in the traffic light, a car that ran a red light, a biker on the road, and a pedestrian crossing the road (Armstrong, 2016). There are clear-cut reactions for these actions. One can hardcode what a car should do when a pedestrian crosses the road in front of it: stop. Moreover, it should always do that. Now, there are edge cases where the car needs to figure out what its priorities would be.

**Combined**

The best of both worlds would be a combination of machine learning and hardcoded artificial intelligence. Most driverless cars operate with a set of hardcoded rules combined with the machine learning for the edge cases (Armstrong, 2016). An evolution of this project would be a project that would have hardcoded standards and guidelines for making points and defending well.

**Video**

**Limitations**

While doing the research, one of the limitations that was quickly observed was a camera limitation. A camera, whether on a smartphone or a dedicated device, has limitations by either the manufacturer artificially or the electronics within it (Mars, 2014). A camera is limited by the many electronics within the device: the image sensor, the image processor, memory, and disk space. Due to these limitations, many consumers check the specifications of a camera or a smartphone with a camera inside before
purchasing it. For example, my Nexus 6P smartphone has the following specifications for video capture: 2160p30, 1080p30, 720p240/120/30 ("Nexus 6P," 2015). Many of these are redundant; therefore, removing the extraneous specifications narrows it to 2160p30, 720p240. This is obviously a shorthand form for what is a complicated set of numbers: the first number refers to the horizontal number of pixels, or points, captured by the camera and the second number refers to the number of times the pixels are cycled through in one second ("High-definition Video," 2005, sec. Technical Details). The p in the middle between the two numbers is not actually a reference to pixels but progressive, which indicates that all the pixels are refreshed that many times.

**Pixels**

To understand this shorthand form, one must understand photos and video. A photo is a still representation of something in the real world captured by a light sensitive material such as an image sensor ("Navigation," 2016). Pixels are reference points in that photo that are of one color at a very certain location in the photograph (Foley & Dam, 1989). When one cycles through photos that are interrelated, the eye interprets the objects on the image as moving. At 24 frames (images) per second (fps), also known as the frame rate, the objects on the screen are seen as continuous motion (Watson, 1986). A higher frame rate produces better picture quality because there is less stuttering.
Figure 1. The picture shows the relative size of the captured area when looking at video size with the same amount of detail in the video. The newest and largest video size, 4K UHD, captures far more details than the existing most detailed format 1080p over an existing area ("4K Resolution").

Stroboscopic Effect

However, after a point, it does not matter how high the framerate is, because the human eye cannot see the difference beyond a point ("Stroboscopic Effect," 2004). However, computer systems can analyze a video frame by frame, so they are not encumbered by the same limitations as our eyes. However, both cameras and the human eye, can produce video that has a stroboscopic effect, a visual phenomenon caused by aliasing (distorting) that occurs when continuous motion is represented by a series of short or instantaneous samples ("Wagon-wheel Effect," 2005). It is also known by other names such as the wagon-wheel or stagecoach-wheel effect. Car wheels are famous for this effect, as even though the wheels are moving forward, people visually see it as the
car wheels rotating backwards. Or less common, but still possible are the car wheels that are completely still. These effects are seen because of too low of a framerate. One can see the images without the stroboscopic effect at a higher framerate than the current ("Wagon-wheel Effect," 2005).

**Figure 3.** In the first image, the viewer sees the first frame and sees the motion as the forward continuous motion of that frame as it is perceived in real life, instant by instant. In the second image, the viewer sees the first frame and sees the motion as the reverse continuous motion of that frame, even though the wheel is still displaying forward continuous motion. ("The Wagon Wheel Effect").

**Figure 2.** The above image shows the definition of frame rate, and what a higher frame rate can do for your video. The 60-fps video has more images per second, allowing it to
capture more detail of the ball’s movements. A higher frame rate also helps to prevent the stroboscopic or wagon-wheel effect ("What Is Frame Rate?", n.d.).

**Project**

How does the frame rate, video size, and the stroboscopic effect influence the project? An increase in either the framerate and video size increases the file size of the video on the hard drive and increases the processing time taken for that video. Thus, finding the lowest possible framerate and video size is recommended while still maintaining the ability to process the video, such as minimizing the stroboscopic effect, which could have adverse effects on the data. The stroboscopic effect does not have as much of an effect of the speed of the ball, considering the ball’s speed would be so great that it would not matter.

The ball’s spin would have an effect. The ball is spinning at an immensely fast number of rotations per minute. When spinning that fast, a ball can spin a few times between two frames of a video. In the PGA tour, a very good driver (golfer) can make the ball spin on an average of 5943 rpm ("What Is Spin Rate?," 2017), which is 99 rotations per second. Putting that into frames can have the ball spinning at approximately 3.5 rotations per frame elapsed for a 30fps video, producing an obvious stroboscopic effect. The camera cannot see the in-between images of the spin of the ball at that frame rate ("Wagon-wheel Effect," 2005). A quick fix for this problem is to increase the framerate of the video, however, it is not as simple as that. Increasing the framerate for the video can decrease the resolution for the video; consequently, the chances for a stroboscopic effect on the spin are decreased, yet the quality of the frame itself for video
processing is also decreased.

**Software**

The library used to build this program is OpenCV (Open Source Computer Vision) on Python. Python is one of the best programming languages for processing data, because of the numerous scientific and mathematical libraries that have been developed for this language (Koepke, 2010). The language itself is not particularly fast (Jaynene, 2008), but the OpenCV Python functions are a mere wrapper for the powerhouse natively compiled functions; this combination uses the simplicity of Python while utilizing the performance of natively compiled code ("How OpenCV-Python Bindings Works?," 2017).

One trick that can speed up the processing is using OpenCV GPU extensions. The GPU (graphics processing unit) has traditionally been used for displaying graphics on the screen, such as rendering a browser webpage or a Microsoft Word document (Buchanan, 2009). However, starting in 2001 and slowly becoming popular over the 1st decade of the 21st century, graphics processors have not only been used for rendering displays; they have also been used for high performance analysis of scientific and graphical data. Even though graphics processors are technically slower than their main general-purpose counterparts, they also have many more cores, or individual units that can run programs (Buchanan, 2009). A general purpose CPU can have up to 32 cores, but they usually only have 2, 4, or 8 cores. The latest graphics processors have 3584 cores with many having at least 50 on the lowest end. My laptop has 512 cores on its Nvidia Quadro M1000M GPU ("NVIDIA Quadro M1000M," 2015). Therefore, even though GPUs are slower, they can
perform greater analysis at a single point in time than a CPU. With the GPU extensions, OpenCV can process images faster.

Python, as mentioned earlier, has many flaws that can slow the project. Whereas GPU extensions are included with OpenCV, they are not included with the OpenCV Python version. Therefore, any OpenCV version in Python does not include the performance enhancing GPU extensions. However, a further iteration of the project could include the GPU extensions if they were natively included in the Python version or if one could port the ball-tracker portion of the program to C++. That port could be simplistic, because of the nature of OpenCV in which the APIs for the functions in any language are similar if not the same.

**Implementation**

**Tripod**

The camera was placed on a top of an eight foot wooden custom-built tripod (shown in Figure 2), attached via a JOBY GorillaPod, a specialized tripod that can be used to attach to metallic surfaces using a strong magnet inside the tripod, unusual rock formations and uneven terrain of all types, metal railings, wood branches and tree trunks. However, I chose to buy a cheaper knock-off brand, which might explain its shoddy construction; the cell phone holder did not properly attach with the rest of the tripod, and the tripod sometimes refused to stay on the proper orientation with the wooden tripod. However, it was still serviceable. The custom built wooden tripod had a long eight foot wooden arm on top of its eight-foot pole, on which the GorillaPod was attached.
Figure 3. This picture shows the custom built eight foot wooden tripod built specifically for this project to facilitate a top-down view of the playing surface to record. The JOBY Gorillapod is on the arm at the top of the tripod.
Problems

There are two major problems with this setup: the tripod itself is not easily acquirable and the camera cannot be easily placed on the tripod. As I mentioned earlier, the tripod is custom-built using wooden pieces. One of the major reasons why it had to be custom built is how much the distance the camera required to be from the playing surface. Not only did the camera have to be away from the action to not interfere with the action, but it also had to capture the entire playing surface to track the movements of the ball. The camera had to be a minimum of five feet away from the playing surface to capture the full view of the playing surface. However, my camera was eight feet away and it only managed to capture only the full playing surface. The view area could be improved with a larger lens such as one on a DSLR, but the amount of improvement would have to be tested.

Another issue is that the camera had to be initiated before or after placing it on the tripod. That is also assuming the JOBY tripod has already been set up on the wooden tripod; if it has not been set up, the JOBY tripod must be set up on the wooden tripod. If the camera is not already recording before setting it up, the record button on the camera had to be pressed, which can be difficult on a smartphone.

If the ball hits the camera, it is not as prone to damage from the ball at this high perch, because most of the ball’s kinetic energy would be dissipated before the ball hits the camera, whereas a side camera might be much more prone to damage from the ball. However, the camera is prone to damage from improper placement and falling from the wooden arm from that improper placement. That is why all precautions were made to be
certain that the GorillaPod and the camera were properly secured on the wooden arm of the tripod.

**Camera**

The camera used in this research was the Sony Exmor IMX377 attached to the Huawei Nexus 6P. The maximum resolution on the camera is 4K UHD at a framerate of 30fps, and the maximum frame rate on this camera is 240fps at 720p resolution ("Nexus 6P," 2015). This was the highest framerate and video resolution available on a smartphone camera, at the time the project was started. Another segment of the camera industry, action cameras, such as a GoPro, could be used if it was not for the fish eye effect and distortion that is a hallmark feature of those cameras (Gibbs, 2015). The fish eye effect is a great filter for a thrilling action video, but not great for a scientific project. When looking at competent cameras that can film at that resolution or frame rate, very few if any at all exist at a price range suitable for most people. One of the cameras that was relatively cheap yet had the features needed was the $899 Sony RX10 Mark IV or the $999 Sony RX10 Mark V (Hession, 2015). However, that is still out of the price range of this project.

**Software**

**Python**

The implementation, as stated in the introduction, is based on Python. OpenCV provides a Python module, which can be easily downloaded similar to a dependency via existing tools such as PyPi (Makai, 2014). The program was split into manageable portions overseen by a master main.py file. The balltracker.py program tracked the ball’s
motion on screen and outputted it into arrays that tracked the ball’s position, 2-D change in position, and velocity. The analyze.py program analyzed the above data points for possible clues about what happened to the ball.

Main

The main.py file takes in the command line arguments, which include the location on the computer for the video file to be analyzed. After parsing through the command line arguments for the location of the video file, it input the file location into the ball tracker and lets the ball tracker retrieve the video, process the footage, and generate a list of positions. After retrieving the list of positions, the main program delivers that list to the analyzer program, which analyzes the list and make recommendations on the player’s abilities and capabilities.

Ball Tracker

Video Capture

The ball-tracker program is in the balltrack.py file. The ball-tracker has an input of the file string. OpenCV uses the VideoCapture function to open the file and start grabbing frames from the video itself. If there are no frames grabbed, the loop breaks and it does not continue processing. If there are frames, the loop starts and continues until there are no more frames. When there are no more frames, the loop sends the list of positions of the ball back to the main program. Inside the loop, the image is being processed very quickly to enhance finding the ball.
Color Conversion

First, the frame is converted from its existing color encoding system to the HSV (Hue, Saturation, Value) system. Usually, the color encoding system used by cameras is the RGB system, or known in OpenCV as BGR, but filtering colors with a scale in RGB can be very hard ("Changing Colorspaces," 2016). The RGB scale is based on a color combination between Red, Green, and Blue; varying amounts of Red, Green, and Blue, all ranging from a scale of 0 to 255, will give you most of the visible colors in the rainbow (Poynton, 2007). However, a similar scale that makes filtering colors very easy is the HSV scale, the Hue, Saturation, and Value scale. (The HSV scale is also known as the HSB or HSL scale for brightness, lightness, or luminosity.) While the RGB scale is seen as a three circle Venn Diagram with the varying colors all embedded in the Venn Diagram, the HSV scale is seen as a cylindrical or conical structure, where the edge of the circle on the bottom is the hue, the depth is seen as the saturation, and the height is seen as the value. HSV makes it easy to filter colors because only the range of colors that match the hue of the object that they are searching for need to be selected; afterwards, the saturation and brightness values only need to be as high or as low as that they do not go into the grey or black hues. When looking at the frame with a 3rd-party image editing software such as GIMP, the range of colors is (22, 30, 63) and (64, 100, 100). However, the HSV scale, unlike the RGB scale, can vary from program to program; the HSV scale for gimp uses H = 0 - 360, S = 0 - 100 and V = 0 - 100, whereas OpenCV uses H: 0 - 180, S: 0 - 255, V: 0 - 255 (K, "Choosing correct HSV values for OpenCV thresholding with
InRangeS," 2012). Thus, conversion between the two systems is a simple multiplication and division problem.

**Image Enhancements**

Second, a Gaussian Blur is applied to the frame to remove small imperfections possibly captured in the image capture during the recording and to normalize the colors across the video. A Gaussian Blur is an equation applied to a pixel on an image (Pi, 2012). After applying a Gaussian Blur, the objects in the frame are then eroded and dilated over two iterations. An erosion is a shrinking of lines and objects and a dilation is a thickening of lines and objects. The Gaussian blur, erosions, and dilations all serve the same purpose: to edit out via a quick and easy algorithm imperfections that would otherwise harm the next algorithm on our list ("Eroding and Dilating," 2014).
Figure 6. These images show the effects of dilations and erosions on this image of a face ("Eroding and Dilating," 2014).
Figure 5. The above image shows what effect a Gaussian blur has on an image, especially with a higher standard deviation ("Motion: Gaussian Blur," n.d.).

**Finding the Ball**

The most important function in this program is the findContours function. This finds the ball using a range of colors specified. Finding the contours simply works in this manner: the range of colors specified becomes white and everything else becomes black (K, "Contours - 1 : Getting Started," 2012). That way it becomes quick and easy to find the ball if there are no other interfering colors that would interfere with the function. A white ball would have major issues with the white stripes on the table tennis top; and any time the white ball passes over those stripes, it would blend with the stripes, possibly making it invisible for a few moments to the function. The orange ball prevents that possibility of a “ghost” ball. If the ball cannot be found, the condition must be that the ball has gone off-screen or is in a player’s hand. The findContours can also help with an imperfectly placed board by helping to situate the ball and the board and find whether the ball has crossed the board.

**Post Processing**

After finding the contours, if any such contours exist, the list of contours must be processed to find the x-coordinate, the y-coordinate, and the radius of the ball. First, the largest contour is found using the Python max function. The enclosing circle for the largest contour is found, in which the function can also find the radius of that circle. Finding the radius gives an approximation of the size of the ball of the screen. To find the position of the ball, we can use the moments function to find the center of the ball. After
finding the center and radius of the ball, we simply append it to the positions list and continue to the next frame. If no contours are found, then [-1,-1,0, -1] is appended to the positions list, because all of those are non-existent values in the range of values in the frame. At the end of the program, the camera and the OpenCV library is released from memory and it returns the positions array for the analyzer program.

Analyzer

The analyzer program is in the analyzer.py file. The analyzer obtains an input of a list of position coordinates from the ball tracker, and quickly finds the instantaneous velocity and acceleration. Instantaneous velocity and speed is calculated via a for-loop that checks if the position does not exist, and if the two positions exist, the instantaneous velocity is calculated, which is the difference between the two positions, and the speed, which is the square root of the addition of the square of the x-vector and y-vector velocities. The instantaneous acceleration is faster, because that is quickly done using the Numpy function diff.

After finding the positions, velocities, and accelerations and compiling lists for each of them, the actual recommendations part is the fun section. All the hard work has been performed in finding the ball on the screen or if it has dropped off-screen. A drop off in velocity could mean it hit the net. A non-increase in radius before going off screen means that the ball never hit the board. And, based on the frequency of these mistakes, the system can quickly identify and recognize the problems that the player frequently encounters.
## Results

Table 1

This is a table for a serve that results in a point in the game. The speed is 3.04 m/s or 6.8 mph. The spin is approximately 150 rotations per minute.

<table>
<thead>
<tr>
<th>Y-Position</th>
<th>X-Position</th>
<th>Radius</th>
<th>Logo Visible</th>
</tr>
</thead>
<tbody>
<tr>
<td>1828</td>
<td>273</td>
<td>80.42744</td>
<td>1</td>
</tr>
<tr>
<td>1635</td>
<td>323</td>
<td>70.36114</td>
<td>1</td>
</tr>
<tr>
<td>1476</td>
<td>366</td>
<td>58.15853</td>
<td>0</td>
</tr>
<tr>
<td>1347</td>
<td>401</td>
<td>47.55879</td>
<td>0</td>
</tr>
<tr>
<td>1269</td>
<td>422</td>
<td>19.14795</td>
<td>0</td>
</tr>
<tr>
<td>1253</td>
<td>425</td>
<td>21.46711</td>
<td>0</td>
</tr>
<tr>
<td>1234</td>
<td>430</td>
<td>23.81283</td>
<td>0</td>
</tr>
<tr>
<td>1209</td>
<td>437</td>
<td>26.93295</td>
<td>0</td>
</tr>
<tr>
<td>1179</td>
<td>446</td>
<td>30.21431</td>
<td>0</td>
</tr>
<tr>
<td>1140</td>
<td>457</td>
<td>33.85709</td>
<td>0</td>
</tr>
<tr>
<td>1094</td>
<td>471</td>
<td>38.01942</td>
<td>0</td>
</tr>
<tr>
<td>1040</td>
<td>487</td>
<td>41.56532</td>
<td>0</td>
</tr>
<tr>
<td>978</td>
<td>504</td>
<td>45.99962</td>
<td>1</td>
</tr>
<tr>
<td>908</td>
<td>522</td>
<td>47.90839</td>
<td>1</td>
</tr>
<tr>
<td>832</td>
<td>544</td>
<td>51.83367</td>
<td>0</td>
</tr>
<tr>
<td>748</td>
<td>567</td>
<td>54.23417</td>
<td>0</td>
</tr>
<tr>
<td>664</td>
<td>587</td>
<td>56.13736</td>
<td>0</td>
</tr>
<tr>
<td>576</td>
<td>609</td>
<td>57.48194</td>
<td>0</td>
</tr>
<tr>
<td>x</td>
<td>y</td>
<td>x_pos</td>
<td>y_pos</td>
</tr>
<tr>
<td>----</td>
<td>----</td>
<td>-------</td>
<td>-------</td>
</tr>
<tr>
<td>493</td>
<td>630</td>
<td>57.6823</td>
<td>0</td>
</tr>
<tr>
<td>412</td>
<td>649</td>
<td>56.00492</td>
<td>0</td>
</tr>
<tr>
<td>340</td>
<td>667</td>
<td>53.46583</td>
<td>0</td>
</tr>
<tr>
<td>276</td>
<td>682</td>
<td>50.32528</td>
<td>0</td>
</tr>
<tr>
<td>223</td>
<td>694</td>
<td>45.99962</td>
<td>0</td>
</tr>
<tr>
<td>178</td>
<td>703</td>
<td>41.43822</td>
<td>0</td>
</tr>
<tr>
<td>145</td>
<td>709</td>
<td>37.53308</td>
<td>0</td>
</tr>
<tr>
<td>121</td>
<td>713</td>
<td>33.56139</td>
<td>0</td>
</tr>
<tr>
<td>107</td>
<td>714</td>
<td>29.22823</td>
<td>0</td>
</tr>
<tr>
<td>100</td>
<td>714</td>
<td>26.39092</td>
<td>0</td>
</tr>
<tr>
<td>100</td>
<td>711</td>
<td>25.4912</td>
<td>0</td>
</tr>
<tr>
<td>106</td>
<td>707</td>
<td>24.64478</td>
<td>0</td>
</tr>
<tr>
<td>117</td>
<td>701</td>
<td>24.68242</td>
<td>0</td>
</tr>
<tr>
<td>131</td>
<td>695</td>
<td>23.51972</td>
<td>0</td>
</tr>
<tr>
<td>149</td>
<td>688</td>
<td>23.28916</td>
<td>0</td>
</tr>
<tr>
<td>169</td>
<td>681</td>
<td>21.66063</td>
<td>1</td>
</tr>
<tr>
<td>125</td>
<td>694</td>
<td>36.05368</td>
<td>1</td>
</tr>
<tr>
<td>50</td>
<td>715</td>
<td>39.11967</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>732</td>
<td>14.935</td>
<td>0</td>
</tr>
</tbody>
</table>

With the spin being shown above being very slow, it is quite possible that a stroboscopic effect could be occurring, with the ball actually spinning at a multiple of 150 rpm, such as 300, 450, or 600. In all of the results, the x and y positions are the most
consistent and reliable data values with the radius being third and the spin being the most unreliable of data points.

**Conclusion**

**Height**

The findings of this project make this solution possible but highly impractical. The table tennis top was placed on the floor of the room and a camera was placed eight feet over the table tennis board. When playing table tennis at its regular height of five feet, the camera would have to be placed at 10.5-11 feet, a height that would be out of reach for most modern one-story homes, and would be possible yet possibly still impractical in two or more story homes, possibly requiring specialized installation that could hurt the aesthetic appeal of a home. The easiest way to obtain this height is to attach it to a staircase or railing on the second floor of the house. Another possible way to fix the height issue, is to use a different camera that can capture more horizontal surface area from the same height, allowing it to be lowered to possibly 9-10 feet, a height possible for most modern homes. Yet, finding such a camera can prove to be costly; most cameras with significantly bigger sensors that can shoot close range with wide angles can prove to be outside of the range of most consumers (Maan & Tharakan, 2015) at a price of nearly a thousand dollars. A wide-angle lens on a DSLR would be the best way to combat the height issue, but it might have an unintended distortion on the image itself (Hildebrandt, 2016). The height issue limits the democratization of this technology.
Details

The second major issue found is that the same camera can have two different viewpoints when trying to record two differing resolutions. The 4K image of Sony Exmor IMX377 was substantially smaller than the image of the 720p from the same camera, requiring the camera to be placed even higher than its eight foot perch. Even though the 4K footage was useless due to its relatively slow frame rate, the footage also became much less useful because of the camera cropping some of the playing surface, which did not occur in the 720p high frame rate footage. Thus, even in the same camera, different resolutions can cause the image to get cropped. Moreover, issues can appear because the 720p footage might not be detailed enough. If the video covers the minimum surface area required, the nine by five foot playing surface must be converted to pixels on the screen. Each pixel must cover 5/64 of an inch or 2 mm. One might think that is small, and it is compared to the relative size of the playing surface. However, if the ball is sitting on the playing surface, that ball is only five pixels wide. Compare that to a 4K video where the ball can be 50 pixels wide or even a 1080p video where the ball can be as small as 25 pixels wide.

Slow Motion Video

The third major issue is that slow-motion video is still considered an exotic feature than a must-have, which is applied more often in action cameras than traditional cameras. Therefore, calculating the spin for advanced processing invites all sorts of problems that have been explained in previous sections with a regular camera (Gibbs, 2015). To mitigate the problem, we have found that most mistakes made by players,
especially beginners and intermediate ones, can be found using the position, velocity, speed, acceleration. However, this does not mitigate the issue that the spin can cause great effects in the direction of the ball and how it moves through the air, especially for advanced players that know how to hit the ball to effect the spin properly. Additionally, to calculate spin properly, high frame rate cameras such as the one attached to the Nexus 6P, iPhone 6S, or the Sony RX100 are a necessity, as exotic and expensive as they are, limiting the accessibility of this program.

**Lighting**

**Current Setup**

The fourth major issue found is lighting. Having a well-lit area is key to a good video, and even areas that could be considered well-lit in real life, on camera were substantially darker and much harder to process, even to the naked eye. Good lighting requires rooms to have a large amount of sunlight or over 7000 lumens, which is the equivalent of over four 100W incandescent light bulbs ("Great Value LED Light Bulbs 14W (100W Equivalent), Daylight, 4-Pack," 2017). Even then, it can seem inadequate for the type of light required for 240fps footage.
Figure 9. This image shows the ball and its logo clearly in the bottom right portion of the image. However, using only a 100W bulb became detrimental as the entire image is not bright. The center of the image has a bright spot while the edges remain quite dark and appear in some areas close to be a very dark grey if not black.

Figure 10. This image shows the ball clearly in a more illuminated room. Compared to the last image which was in a room with only a 1500 lumen light bulb, this image now
has 7500 lumens of light. The difference is clear; this image is much brighter and sharper than the one before it.

**Exposure**

Even with the bright light, the 240fps footage is grainy and causes major processing problems, because the camera increases the ISO. The ISO refers to the light sensitivity of the light sensor; higher ISO images are usually more grainy, more blurry, less sharp, and less detailed than their lower ISO counterparts. A lower ISO image also has better color reproduction and range. However, a lower ISO also means a lower exposure, which can mean a darker image than the higher ISO image (Mathies, 2017).

Therefore, usually in low-lighting or inadequate lighting situations, photographers must find the right balance between the ISO and the brightness of the image.

Another component of exposure is the shutter speed. Shutter speed is how much time the image sensor is exposed to the picture. As frame rate increases, the shutter speed usually increases, but that is not necessarily true for the opposite direction. One can have normal frame rate video with a very low shutter speed. Lowering the shutter speed also reduces motion blur but also increases the need for a higher ISO. The motion blur noticed in some of the footage can be attributed to a shutter speed that is too low; however, fixing that might require more ISO or cause the image to be darker, two trade-offs that might be even more detrimental to the footage.

The graininess of the footage makes it difficult to process the spin of the ball, because dark patches interfere with finding the logo. Also, another major issue is that a logo can get easily be blurred out of the image because of the graininess; graininess in
footage indicates a lack of sharpness. The lack of proper exposure also hurts the data gathered.

Figure 7. Turning the ISO slowly up on a camera also increases the blurriness and graininess of the image itself. Whereas the numbers on the tape measure are clearly visible in the ISO 100 image, the picture becomes grainy in the ISO 1600 to the point where the numbers are barely readable, if readable at all (Goldman, n.d.).
Figure 11. This image shows the previous image at 240fps rather than the usual 60fps. The image is clearly grainy and blurry due to the large ISO, impeding the processing and possibly blurring out the logo. This can have a negative effect on the spin data, because it can make it more difficult if not impossible to see the logo for the ball if it is blurred out.

Alternatives

Bright sunlight seems to be the best way to record footage, but that limits the amount of time that one can use this system and further limits the setting of the system. However, the photons must still reflect off the surface of the board and come to the small camera sensor of the Nexus 6P. It is possible that a camera such as the RX100 IV can help with the lighting by having a larger image sensor such as the one-inch Exmor RS sensor unlike the 7.81 mm Exmor sensor in the Nexus 6P (Moynihan, 2015), which means the former is 3.25x larger than the latter.

Alterations

There are many alterations that could be made to help better detect the spin. Rather than trying to detect a logo on the ball, if a multi-colored ball could be produced, that could help better detect the spin. Another alteration could be another camera placed at the side, which could help the camera not increase the ISO during high-speed frame rate capture and maintain a better hold on image quality. A camera not facing downward vertically can only capture a certain amount of light, which is significantly less than one facing upward or sideways due to the sensor’s positioning relative to the sun. Another alteration to the setup could be the placement of two cameras much closer to the playing surface. Rather than having one camera capture a 9x5 playing surface, it would
only have to capture a 4.5x5 playing surface. However, it would cause issues of stitching together the videos in processing. Furthermore, it could causes issues of shadows in the images; the current setup is above or at the same range of most of the lighting in the room. A lower setup could cause major shadows over the playing surface, and to avoid that, lighting would have to be attached to the tripod. There is a possibility of using a technology such as LIDAR; however, LIDAR is unable to capture the spin of the ball.

**Better Camera**

In addition, the issues discussed above, including the lighting, the exoticness of high frame rate, and the height impracticality, possibly can be solved by a camera such as the last-generation RX100, which retails at a price of a minimum of nearly a thousand dollars on Amazon or BHPhoto (Hession, 2015). Many will raise the questions of “If this requires a huge upfront cost, would it be cheaper to just hire someone to coach a ping-pong player? Would this program actually democratize the sport of table tennis?” After doing this project, the answer to that question is yes, to some extent, it will. Many middle schools and high schools, who might not have otherwise sponsored a table tennis team, due to lack of funding for a coach, might spring for this type of system. Many colleges could pay for this type of system in addition to a coach, because not only does this system provide coaching, but it also provides data that human coaches can use in coaching their players.
Figure 8. As image sensor size increases, more of the image appears from the same perspective. A smartphone image sensor has a much smaller viewpoint than the one of the full frame DSLR. Increasing the size of the image sensor inside can help reduce the height needed to view the full playing surface (Crisp, 2013).

Time

Nevertheless, the ultimate solution might not be any of the above. The solution might simply be to wait for more time to pass. Time has proven to be a great ally in solutions. Less than three years ago, this project might have been considered out of the realm of possibility. The iPhone 6 had not launched yet with 240 fps slow motion video capture ("iPhone 6," 2011). Even most high-end DSLRs did not have slow motion capture; and slow motion video capture might only have been in the realm of budget professional video cameras, easily costing over two thousand dollars. Additionally, as smartphone cameras have improved in the past to challenge to remove the compact camera market and even possibly challenge budget DSLRs (Moynihan, 2015), they could continue to improve. Sony’s newest smartphone camera sensor can obtain video at 1000
fps at 1080p (Singleton, 2017), and their latest smartphone, the Xperia XZ, can obtain video at 960 fps at 720p (Kelion, 2017). Hopefully, in the future, this technology is not limited by optimal lighting conditions, or at least lower levels (120-240fps) of frame rate will see a drastic improvement in video quality in low lighting conditions. Or, even better, low end DSLRs and video cameras could adopt this feature as a method to differentiate themselves from smartphones.

Concluding Thoughts

The program seems ready to take on ping-pong and help coach a new generation of ping-pong players. However, the perfect environment and a high-end camera are sadly a necessity; for now, it seems that only time or a large amount of money will solve that issue.
References


How to Play Ping Pong (Table Tennis). (2017, April 17). Retrieved April 21, 2017, from http://www.wikihow.com/Play-Ping-Pong-(Table-Tennis)


Maan, L., & Tharakan, A. G. (2015, October 29). GoPro faces tough competition as


THE PING-PONG PROJECT

https://en.wikipedia.org/wiki/Nexus_6P


doi:10.1080/02640414.2013.805885