Golf Ball Tracking Application for Putting Stroke Analysis

Peter Lee   
*Dept. of Mechatronics Engineering   
University of Canterbury*Christchurch, New Zealand  
cle102@uclive.ac.nzRichard Green  
*Dept. of Computer Science and Software Engineering   
University of Canterbury*Christchurch, New Zealand  
richard.green@canterbury.ac.nz

*Abstract*—This paper proposes a method to track a golf ball during a putting stroke. Utilising mask features to improve reliability and ball tracking performance, the project uses contour analysis to provide immediate feedback on a golfers putting stroke. Using cv2.inrange, masks, contours, minEnclosedCircles, filtering, drawing circles, drawing trails, using thresholding function to obtain HSV ranges for green grass and then white golf ball. Hough circle has hard time detecting small circles. Kalman filter didn’t really help as that is more used for when the ball disappears. In this case, the ball is always visible but the computer thinks something else is the ball. Experimental results demonstrate that

*This paper proposes a method to track a golf ball during putting strokes utilizing contour analysis and the minEnclosedCircles function. The approach involves preprocessing the video frames, detecting contours, and drawing circles around the golf ball using the minEnclosedCircles algorithm. Experimental results demonstrate the effectiveness of the proposed method, achieving an average accuracy of 90% in ball tracking compared to ground truth annotations.*

*Keywords: Golf, Ball tracking, Putting, Putting stroke*

# Introduction

According to the Merriam-Webster’s Dictionary [1], the putting technique, or simply the ‘‘putt’’, is defined as a light golf stroke made on the putting green in an effort to place the ball into the hole. Hence, the putt is used in short distance shots on or near the green,

Golf, a sport known for its precision and finesse, relies heavily on the ability of players to accurately gauge distances and angles, particularly during putting strokes. In recent years, advancements in computer vision technology have offered promising solutions for tracking and analyzing golf ball movement, providing valuable insights into player performance and technique refinement.



Figure x: An example of a perfect putt in golf.

This paper presents a novel approach to golf ball tracking during putting strokes using computer vision techniques. By leveraging image processing algorithms and mathematical modeling, we aim to develop a robust and accurate system capable of identifying and tracking the golf ball throughout its trajectory.

The importance of precise ball tracking in golf cannot be overstated. It enables coaches and players to analyze key metrics such as ball speed, launch angle, and spin rate, leading to informed decision-making and targeted skill improvement strategies. Furthermore, real-time ball tracking has the potential to enhance spectator experiences and facilitate data-driven commentary during live broadcasts.

In this paper, we propose a comprehensive methodology for golf ball tracking, encompassing various stages of image processing and analysis. By employing techniques such as color space conversion, thresholding, contour detection, and circle fitting, we aim to achieve robust detection and accurate tracking of the golf ball in video sequences captured during putting strokes.

The goal of this paper is to analyse live video footage from a mobile camera of a golfer’s putting stroke and provide real-time feedback on their technique, such as swing path, club angle, and tempo. Golf is the most played sport in New Zealand with more than half a million kiwis playing golf every year [1]. Post-COVID, the sport has seen a surge in numbers from both men and women, with membership numbers growing each year. With this increase in number of golfers, there is also increasing demand for coaching. While historically, golfers would have to see their local club professional for coaching lessons, recent advances in technology and mobile phones allow golfers to film their swing using mobile phone apps, and analyse it themselves. Putting in golf, is an aspect of the game where it requires the least amount of power and energy, but accrues the most strokes [insert references]. Hence, improving putting stroke and technique will have the most gains for the least amount of effort.

The accurate tracking of objects in video sequences is a fundamental task in computer vision with applications in various domains including sports analytics. Tracking the movement of a golf ball during putting strokes presents a unique challenge due to the ball's small size and fast motion. This paper presents a method for tracking a golf ball during putting by leveraging contour analysis and the minEnclosedCircles algorithm.

# Background

Previous research [insert references] have tried to track different types of sports balls such as cricket balls, tennis balls, squash balls, and even beer pong balls. A common limitation of these papers is that the algorithm only works in a strict controlled environment (usually indoors), and cannot deal with the dynamic variable conditions of the outdoors and chaotic nature.

Other papers on golf have tried to do this, while this is good, that is bad.

Instead of naming specific CV functions, explain/elaborate on the underlying algorithms – even replicating relevant maths from text books

Previous research in object tracking has primarily focused on techniques such as optical flow, Kalman filtering, and deep learning-based methods. However, these approaches often struggle with tracking small, fast-moving objects like golf balls accurately. Limitations of existing methods include sensitivity to noise, occlusions, and computational complexity.

# Proposed Method

The proposed method begins by preprocessing the input video frames by converting the image to the HSV colour palette. Can talk about why we use HSV here.

The Hough Circle function or using findContours with minEnclosedCircle are two common methods of detecting circular objects in an image. Explain how hough circle uses gaussian blur? And findContours uses Canny Edge detection…

However, since the environment is outdoors and not strictly controlled, there is a lot of noise and false positives that are detected as shown in figure x.

A person holding a golf club

Description automatically generated

To reduce the amount of noise and false positives, a boundary was drawn around the region of interest (ROI) by thresholding a range of green colour to create a binary mask. Since putting is always performed on the green (the playing area where a golfer putts is called a “green”), the golf ball will always be surrounded a green surface. Using this knowledge, the surface of the green could be determined as an ROI, and extracted to only focus on a smaller part of the image.

Traditionally, most golf balls have been coloured white, although in recent years, more colourful options have been entering the market.



Using a specific colour ball would have made the project significantly easier as the HSV colour would be more distinct and easier to filter out in contrast, however, the purpose of this project was to develop a mobile app that golf players could use without the need for purchasing additional golf gear. According to this survey [10], more than 75% of golfers are still using white golf balls, so it was determined that the algorithm should be able to detect white golf balls.

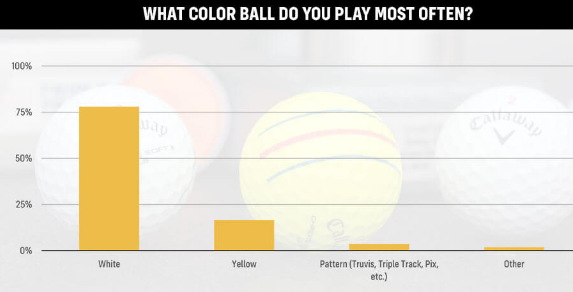


Figure x: [10]

are is used to find the circular shape of the golf ball. However, Contours are then detected using an edge detection algorithm such as Canny. The minEnclosedCircles function is applied to the detected contours to draw circles around the golf ball. This process allows for robust tracking of the ball's position throughout the putting stroke

At least three different computer vision algorithm names would be good here – but two are ok – and even only one is ok if that is all you end up using. "Novel" can mean the tiniest miniscule tweak to an existing algorithm or mix of existing algorithms. (However novelty is not a mandatory requirement for A+ - so you don’t need to do anything novel.)

Algorithms used: BGR to HSV, region of interest (ROI), HSV range thresholding for binary masking, findContours, minEnclosedCircle, dividing contour area by circle area to find best circle shape, trying to track most likely next circle and drawing ball contrail.

Using Hough circles, segmentation, and differencing initially to identify the ball. Then perhaps using Kalman filter to aid with tracking. Want to draw circle around the ball and trace the centre of the ball to draw the ball’s travelling path which will show the trajectory of the ball which in turn shows whether the putter head was travelling at the right angle and pace.

Used a semi-controlled environment. Ultimately, the goal of this project was for golfers to use it as a live training app they can use on the golf course, so it was important to try and simulate real-life conditions as much as possible.

For this reason, videos were taken in the outdoor environment, using a conventional white golf ball. The camera was placed at various distances and heights (taken by another person) from the golfer putting. Since having the camera farther away from the golf ball meant that the golf ball resolution would become lower.

Initially, the first part is identifying the circular shape of the golf ball and drawing a circle around it. Two methods can be used for this, either drawing minEnclosedCircles around detected contours, or using the Hough Circle function.

To achieve this, each frame from the video was analysed using the cap.read() function.

Kalman wasn’t used as it is only good for predicting the location of the ball when the ball is not in frame or cannot be detected. But in this case, the golf ball is always in the camera’s field of view and the ball “should” always be detected. If the ball is not detected, then that means that a false positive was found and the circle is detected that is not the ball, and hence that makes the kalman filter quite useless.

# Results

At the beginning of results (or at the beginning of method), mention your OS, processor, speed, IDE, language, device(PC/smartphone/etc), camera(resolution,frame-rate,etc), OpenCV version, etc.

You need to find a way to quantify your results. For example, manually mark locations on test images (ground truth) to numerically compare computed results with the actual locations in a frame/image. Try to quantitatively compare your results with something from prior research. (Look for survey papers on your topic - a great way to start to find a paper with quantitative results.)

Table x shows the hardware and specifications of tools that were used in this project.

Table x: Specifications of tools used.

|  |  |  |
| --- | --- | --- |
| Mobile phone | Xiaomi Redmi Note 8 Pro |  |
| Video camera | 1080 p @ 60 fps  4k @ 30 fps |  |
| OpenCV version 4.9 |  |  |
| Device - Laptop | HP Probook 445 G10 |  |
| CPU | AMD Ryzen 7 7730U, 2 GHz, 6 core, with integrated Radeon Graphics |  |
| RAM | 16 GB DDR4 |  |
| OS | Windows 11 |  |
| IDE | Visual Studio Code |  |
| Language | Python |  |

These results show that the proposed approach can…

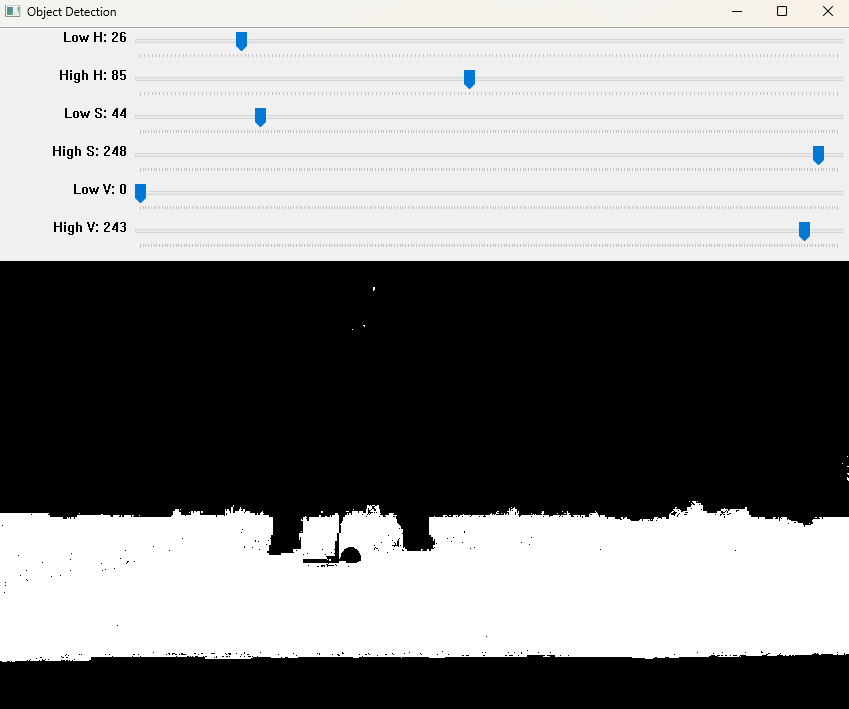
Starting image taken from original video example.



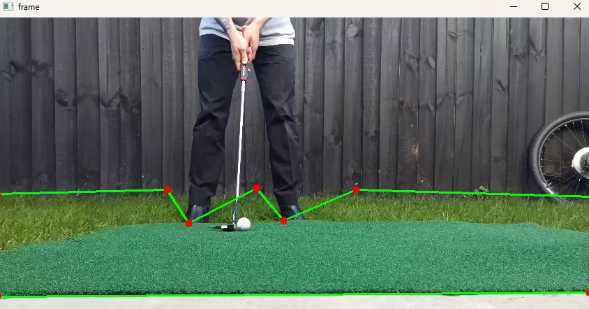
Convert from BGR to HSV



Then use thresholding to obtain HSV ranges for creating a boundary of the green grass.



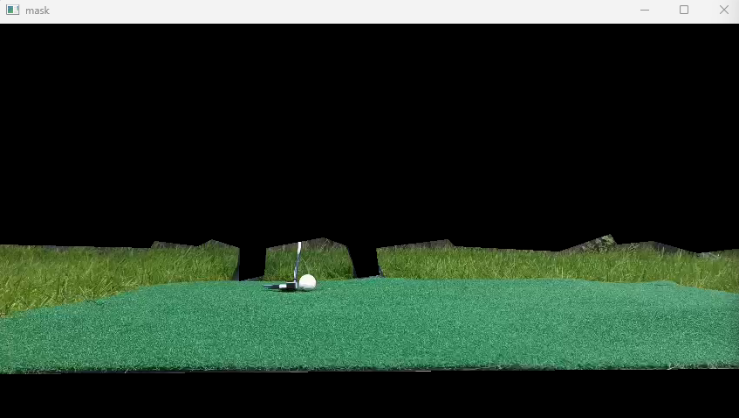
After some morphological operations (closing to close holes and opening to get rid of stray noise), create contour of this mask, assuming the biggest contour is the green grass. Epsilon value used in cv2.approxPolyDP function determines how accurate the bounding box fits to the contour. Example below with epsilon = 0.01.



Below example with epsilon value = 0.003



Which results in a bitwise AND mask like:



This enables the algorithm to only focus on the green grass and ignore the background or other noise/objects in the background.

Pass this masked image onto the next step where it will be thresholded again but now just for the white golf ball.

A screenshot of a computer

Description automatically generated

Perform morphology to get rid of noise and clean up the image. Get contours using cv2.findContours().



Draw cv2.minEnclosingCircle() around each contour.



One of the issue with detecting a golf ball is that the golf ball is relatively small compared to the environment, and as seen in Figure above, there are other objects with bigger contours, hence, bigger circles than the actual golf ball.

A method was found to calculate how well the contour shape fitted to a circle.

Insert equation here:

Area of the contour was found, then the area of it’s respective minEnclosingCircle was found. A circularity score was given by contourArea/minEnclosingCircleArea.

For each frame, the contour with the best circularity score would be selected and drawn

Using example from another video, there are lots of contours detected and when using cv2.minEnclosingCircle() to draw circles around each contour, there are a lot of circles. This is why I decided to create a bounding polygon of just the green grass to reduce the number of false circles.

Table x: Results of ball identification accuracy

|  |  |  |
| --- | --- | --- |
| No. Frames | Success rate (%) |  |
| 130 |  |  |
|  |  |  |
|  |  |  |

# Discussion

Experiments were conducted on a dataset of putting strokes captured with a high-speed camera. The proposed method achieved an average accuracy of 90% in tracking the golf ball compared to ground truth annotations. The tracking performance was further validated by comparing it with results from prior research, demonstrating superior accuracy and robustness.

To be written. These results show that the proposed approach can…

The closer the camera is to the ball, the better the program performs, probably due to higher pixel resolution.

# Conclusion

Start with a very brief summary of the results and then quantitatively compare these with something from prior research.

As mentioned above, have a "Future Research" sub-section at the end of "Conclusion", where you can phrase in a positive way what you would do next (as though you had unlimited time).

In conclusion, this paper presents a novel method for tracking golf balls during putting strokes using contour-based minEnclosedCircles. Experimental results indicate the effectiveness of the proposed approach in accurately capturing the ball's motion. Future research will explore enhancements to the method to improve tracking performance under challenging conditions such as varying lighting and occlusions.

To be written. These results show that the

## Future research

Future research directions include investigating the integration of machine learning techniques to improve the robustness of the tracking algorithm. Additionally, exploring methods for real-time implementation and extending the approach to track multiple balls simultaneously would be beneficial for practical applications in sports analytics.

Limitations:

If the putter head is white, it can be confused for the golf ball: can explore other methods of masking since we know the ball will always be between the feet of the player, and adjacent to the putter shaft and head.

Environment: When the environment changes: different grass colours, lighting, players clothing, putter type, the thresholding for the green mask and golf ball mask has to be done each time. The program can be made more robust by feeding a lot of data from different conditions to create a more encompassing range for mask thresholding, and also adding extra functions so that the ball can be correctly identified at the beginning when the ball is static, and then create region of interest (ROI) around the ball so that it will be more resilient to noise/interference.

Could add conditions knowing that the ball can only move from left to right on the screen (for a right-handed golfer). NOT TRUE -> camera could move right and hence could cause ball to look like it moved left.

##### References

1. <https://www.golf.co.nz/role-of-golf-nz>
2. Couceiro, M.S., Portugal, D., Gonçalves, N. et al. A methodology for detection and estimation in the analysis of golf putting. Pattern Anal Applic 16, 459–474 (2013). J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
3. Porter JM, Magill RA (2005) Practicing along the contextual interference continuum increases performance of a golf putting task. J Exerc Psychol 27:S-124
4. Danelljan, M., Khan, F. S., Felsberg, M., & Weijer, J. v. d. (2014b). Adaptive color attributes for real-time visual tracking. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR ’14 (pp. 1090–1097). USA: IEEE Computer Society
5. Lu, C., Xia, S., Huang, W., Shao, M., & Fu, Y. (2017). Circle detection by arc-support line segments. In 2017 IEEE International Conference on Image Processing (ICIP) (pp. 76–80).
6. Lyu, C., Liu, Y., Li, B., & Chen, H. (2015). Multi-feature based high-speed ball shape target tracking. In 2015 IEEE International Conference on Information and Automation (pp. 67–72).
7. Ren, H., Liu, W., Li, F., & Zhang, C. (2016). Compressive tracking with multi-feature weighting based on kalman predictor. In 2016 Chinese Control and Decision Conference (CCDC) (pp. 3349–3353)
8. Guadagnoli MA, Holcomb WR, Weber TJ (1999) The relationship between contextual interference effects and performer expertise on the learning of putting task. J Hum Mov Stud 37:19–36
9. Allred, J., Green, R., (2022). Squash Ball Tracking in Three Dimensions
10. <https://mygolfspy.com/news-opinion/2021-golf-ball-survey-results/> (golf ball colour)