Benjamin Kelly

GEOG 398

**Final Project Report**

Introduction of Goals, and Problem Statement

The problem attempting to be a dressed with this project is the lack of a comprehensive application to track to players movement namely during soccer match so that this data, and information could be used to display the position of players on the field, at any given time. This project will mainly be looking to take in game footage, and display the location of players, and the ball as a top down, or over head view. The outcome of this project could help coach, and players better understand the game, and were they could improve. The application could also give people who only have access to a smart phone the ability to watch the game in real, however from the previously state Overhead view angle.

Literature Review on Topic

The inspiration of this project came from a paper titled “A Computer Vision Based Web Application for tracking soccer players” authored by; J.M.F. Rodrigues, P.J.S. Cardoso, T. Vilas, S. Bruno, P. Rodrigues, A. Belguinha, and C. Gomes. In this paper the author describes the methods, and steps they used to track soccer players, and display their positions on a field so that the data could be viewed from a top-down perspective. Many of the methods, and techniques used were foreign to me when I started the project. This required me to do a fair amount of reading, and research to become acquainted with these new terms. Another paper that I used was titled ‘Transformation Technique’ authored by M. Venkatesh, and P. Vijayakumar. This paper mainly discusses how an image might be manipulated to produce the same image but from a different perspective. Going into depth on the elements of coordinated shifting, and projection I used this paper to help me understand how I would shift the images in the video so that the filtering, and other methods learned in the first paper mentioned would have the greatest, and most accurate affects. In combination with the two papers, I also used the Open-CV library documentation consistently in my project. Many of the methods, and functions were well defined which helped me tremendously. As the python Open-CV library is very large, and I would have likely be lost without the documentation.

Data used, and potential limitations

Originally for this project I hoped that I would be able to take in any type of game footage to be used to a top-down view of any game. However, as my project pushed toward, I saw the inherent issues with this goal. For example, I saw that most game footage have replays, and zoomed in shoots that would disrupt my program. My program also not having a way to track player movements that are not in the frame of the video. Meant that if players are not in view they would be counted as not being there. For this reason, I need to standardize what type of video should, and could be used for this project. I found that panoramic match footage would give me all of the elements I need to produce a top-down view. Panoramic footage is video that has the whole view of the field, and players without having the camera pan, or move.

The video/data I used for the Panoramic view of the soccer match came from YouTube, under a video titled “Panoramic video of the football pitch” by PANORIS.



Figure 1: Example of Panoramic Soccer Match Footage (VisionCam)

While this might not be video that your average camera could capture there are companies that rent out such equipment. I am fairly rooted in soccer, and have been playing my whole live. As of recently I have seen cameras recording panoramic soccer match footage at as low a level as U-10 (10-year-old players). Meaning that even small clubs, and team would be able to capture the needed footage, and use this application to better understand the positioning of each player. The company that I have seen used the most is called “Veo”, and the camera system they have developed fairly simple, and could be operated by your average parent while also watching the game. The camera is made up two separate cameras each taking in half of the field, and then using some proprietary software both of these video streams are stitched together to make the panoramic video that my application needs to function properly.

Outline of Methods

The main methods that I used for this project can be broken down into two type. The first being NumPy which mainly gave use the array, and data structures needed for the Open-CV library to work properly. Simply NumPy is a package focused on scientific computing in Python. The library provides multidimensional array objects, as well as another derived object. We use these objects to hold the data for each frame of the video. Having the video held within a python NumPy array means that we are able to manipulate, and extract data from it. There are libraries such as Dask that make use of NumPy array but structure them in such a way that they are more gear to working with larger datasets during the course of this project I would have like to include some of these methods. However, time did not permit me to overhaul the main body of my application to include these new features.

The second main type of methods that we use come from the Open-CV library. Open-CV is python library that focuses on Computer Vision. Open-CV was originally developed by the Intel Research Initiative in 1999. I used the Open-CV library, and its method many times in my application. For example, the blob detection used to identify players on the pitch, and ball is an Open-CV function that was critical to the success of my project.

Application Data flow of my project/application:

1. Take in match footage
2. Cut footage on the corners of the field
3. Transform this cut to be a top-down view of the field

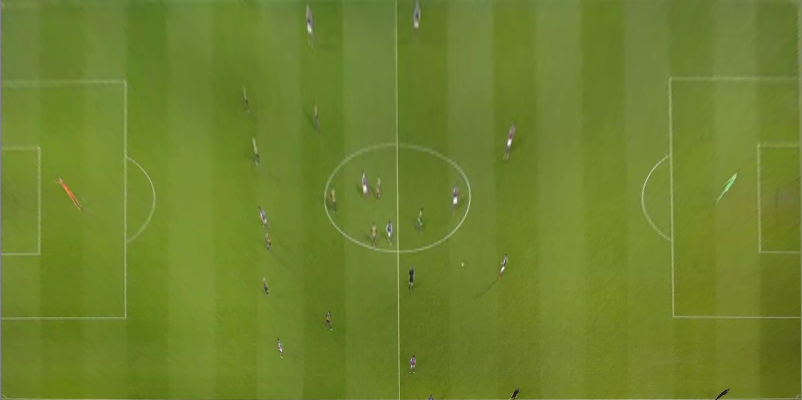


Figure 2: Top-Down view of a Soccer Match

1. Process the incoming using HSV, and other filters to produce a bit image (Black, and White)
2. Take in this processed video, and use Open-CV blob detection to find the location of the players, and the ball.

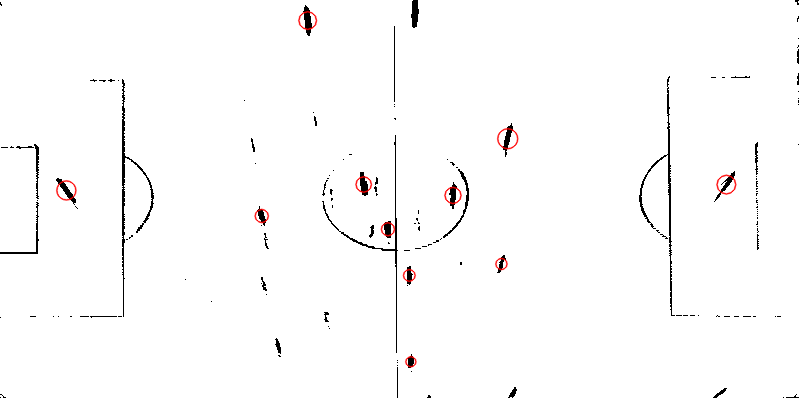


Figure 3: Processed Image with Players detected

1. Returning the players, and ball location (x, y) from the blob detection
2. Plot this (x, y) coordinates on a new field, and display it

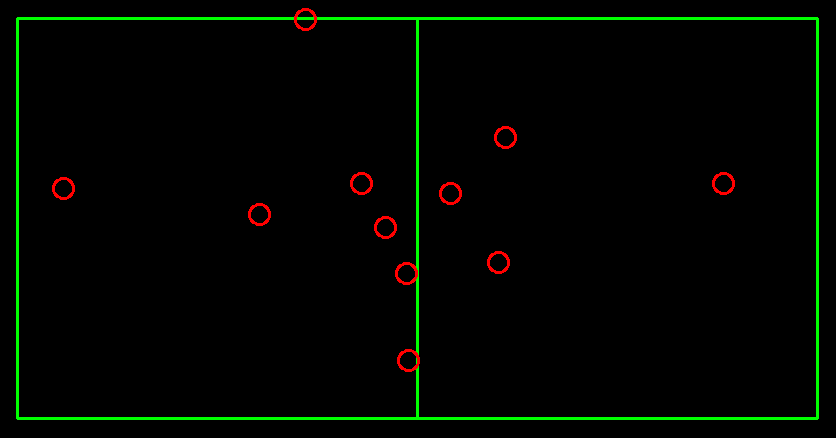


Figure 4:Final Return of the Program

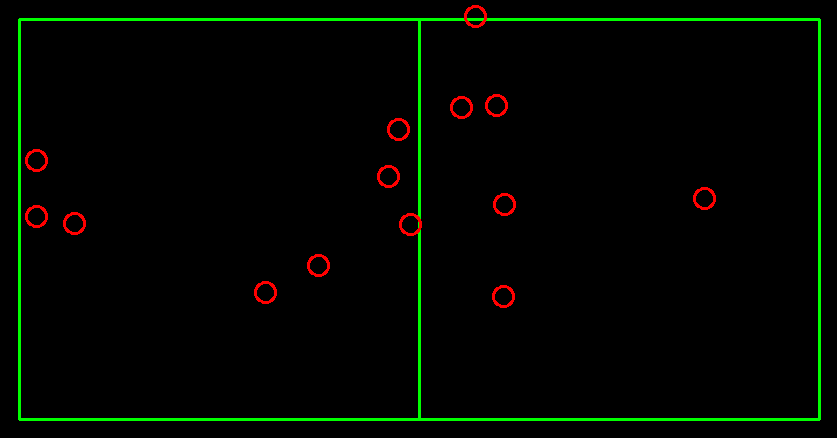
1. (When the video ends the application end)

Some of the main functions that I used in my where;

* Cv.Event\_LButtonBLCLK: This function helped to cut the field at the corner
* Cv.SimpleBlobDetector: Class used to detect players, and the ball on the field
* Cv.CvtColor, and Cv.inRange: Both of these functions where used to produce the bit image of the field
* Cv.getPerspectiveTransform: This function was used to produce the top-down view of our original video
* Cv.WarpPerspective: This function was also need to produce the top-down view of the video
* Cv.VideoCapture: Used to take in the video, and create a NumPy array from it

Analysis, and Results

The result on the project was rough working copy of the application capable of. Overall, the outcome of the project was successful. So far as I was able to achieve many of the main goals that I original set out to solve. Those goals being take in soccer match footage, and produce a top-down view of the game. However, give more time with this project to optimize detection, work on a pleasing user-interface, and improve effective I believe that this could be a great resource to help even small youth soccer clubs/teams to approach the game from a more analytical perspective. As my project was more focused on a tangible product than a statistical analysis. I little in the way of hard data to support my findings but from looking over the video, and results of the application in great detail I have come up with some analysis of my own. Firstly, the blob detection parameter needs to be more defined in order to select only the players on the field. Currently my program some time mistakes lines, or other parts of the field as a player or the ball.



For example, in the Final Return above we can two points that have been identified as players (Circled in Blue). These points are actually just parts of the field. As a result of the filter, or blob detection it they are seen as players because of parameters that do not fit every edge case. I am not sure how I would correct this problem for all edge cases as if I make the parameters to strict the program will reject some players as just part of the field, and if I make then to open, we increase the likelihood that parts of the field will be as players.

At its current level of accuracy, the program would not be effective at delivering insights, and data that could be used to make real time decision during a soccer match. In addition, I need to find a way to determine which team a player is apart of. This would have to do with the color of their shirts however making the program aware of these differences, and what they mean in terms of the game is a challenge that must be over come for the application to be useable in a real-world scenario.

The main source of data I have to determine where time, and effort would be used to the greatest effect is my own experience with soccer, and programming. As far as the function of the application I know that for a soccer team to gain the greatest utility out of the application it needs to be able to function on a live video stream, and the final out should be viewable by players or coaches on their phones. There also needs to be a way to produce a heatmap of each player/team. So that a coach can make tactical decisions in real-time. In combination with all of these additional features I think the most of the code could be optimized to improve computational complexity. Currently while the program is running its utilizing about 30% of CPU powers to run.

My computer has a Ryzen 7 2700 (8-cores), and I do believe that this program would much more taxing on a small less powerful computer, and maybe make it impossible to run. As during a low-level soccer match it would be impossible, or at least impractical to bring a large, and powerful computer to run this program. The program would need to be able to run on a Raspberry Pi powered by a battery.

Conclusion

To summarize the goals of this project was to build an application to take in soccer match footage, and return a top-down view of the game. So that people would be able to watch the game on their smart phone in real time. The application would fill in the gap left open by soccer app who only have a system to track a game play by play if an actually person is watching the game, and putting in the actions in manually. The project achieved the initial set; however, the application still needs updates, and new features for it to be of use to our average soccer player or coach who I would not expect to sit down, and try to understand what the code is actually doing. There still needs to be a useful user interface as well as other features added before I would say that this application could be use by the public as open-source software.

In closing I really did enjoy working on this project as it allowed me to combine two of my great passion for soccer, and for programming. During the course of this project, I was able to identify a problem, and find my own way of solving it. I feel that I work best when I am given a defined task but allow to solve it in my own way. In some sense it is more of a creative exercise trying to make each element of the code, and project flow into one another. I hope to continue to work on this project in the future, and maybe after some time, and effort it can be of use to someone, or least myself.

During this project I was given the opportunity to work with the Open-CV, and computer vision more generally. This was a great experience for me as computer vision seem like a field that will have a lot of growth in the future as more, and systems incorporate video it their data flows. Self-driving cars, robotic, and sport analytics are all growing fields that would benefit greatly from improved computer vision technology. I hope that I will be able to continue working on interesting projects like these in the future as I transition to the workforce.

**References**

Rodrigues J. et al. (2014) A Computer Vision Based Web Application for Tracking Soccer Players. In: Stephanidis C., Antona M. (eds) Universal Access in Human-Computer Interaction. Design and Development Methods for Universal Access. UAHCI 2014. Lecture Notes in Computer Science, vol 8513. Springer, Cham. <https://doi.org/10.1007/978-3-319-07437-5_43>

Venkatesh, M., &amp; Vijayakumar, P. (2012). Transformation Technique. International Journal of Scientific &amp; Engineering Research, 3(5).

OpenCV modules. (n.d.). Retrieved December 15, 2020, from https://docs.opencv.org/master/

NumPy v1.19 Manual. (n.d.). Retrieved December 15, 2020, from <https://numpy.org/doc/1.19/>

**Data Used**

VisionCam. “Panoramic Video of the Football Pitch.” *YouTube*, YouTube, 3 Jan. 2016, www.youtube.com/watch?v=HBTFZwMdcCw.