Augmented Reality First Down Line Drawing for American Football Broadcasts

Ryan Baker
Department of Computer Science
Golisano College of Computing and Information Sciences
Rochester Institute of Technology
Rochester, NY 14586
rwb4048@rit.edu

***Abstract*—This paper discusses a simple, cost-effective, plug-and-play method to recreate the yellow first down line effect seen in many professional American football broadcasts. The virtual effect is intended to span an American football field during play, denoting where the offense needs to reach for a first down. This effect accounts for moving and zooming cameras as well as ensuring the virtual line is not drawn ontop of any players or objects in the field of play. Unlike the NFL's version, this method does not require any specialized equipment or predetermined measurements to create the desired effect. We have found the methodology outlined in this paper to be an effective way to add an augmented reality first down line to a football broadcast. In addition to this, this method can be modified to include additional effects to the field, such as the game score or virtual advertisements that users may want to add to their broadcasts.**

I. Introduction / Overview

American football is the most popular sport in the United States, with hundreds of millions of viewers tuning in every week during the football season. One of the most notable components of a football broadcast is the bright yellow first down line that is drawn on the field. In order to obtain the effect of the first down line, the NFL and the television stations that broadcast the games need a lot of expensive equipment to get the desired result. The necessary technology includes an exact 3D model of the field of play, three or four specialized cameras pointed at the field, specialized camera mounts that track each camera's movement, and a high-end computer for each camera to process it's feed [1]. The problem with the system that the NFL uses is that it requires all this overhead to use and is therefore not usable in different situations where this specialized technology is not available. Broadcasts of lower-level games, such as local high school games or division three college football games will not have the first down line present in the broadcast since they do not have access to the technology that the NFL has.

In this paper we will present a much more lightweight plug-and-play way to add the yellow first down line to any football broadcast. The system we use will be able to draw the first down line only using a camera and a computer for the purpose of post-processing. No additional specialized equipment will be needed. Users will be able to enter where they want the yellow line to be drawn on the field through a user interface, and after pointing the camera at a football field, this system will be able to identify the field and find the correct yard line to draw on top of. The line will only be drawn on the field and will use background removal techniques to ensure that it is not drawn on top of any objects or players on the field. This system is designed to be used with any camera with reasonable quality as well as work well on computers with mid to high end hardware. For this reason, this technique is a great way for lower-budget football games to include the first down line in the broadcasts of their games.

For the purposes of this paper and the experiment done in this paper, a desktop-sized replica of a football field will be used to simulate the real thing. In the experiment, it is shown that this system is able to accurately place the first down line in 83% to 88% of frames. Additionally, the system is able to get this high accuracy with minimal delay running at about 20 frames per second with only a couple tenths of a second delay on average hardware.

II. Related Work

Since this paper's purpose is to recreate a well known effect used in NFL broadcasts there is plenty of related works. First, we will go over what is done by the NFL. The name of the technology used by the NFL is titled ”1st & Ten”, developed by the SportVision company. The system uses three different cameras placed along the field, each with electronic encoders that monitor the pan, tilt, zoom, and focus of each camera [1]. In addition to this specialized equipment, professional football broadcasters take very precise measurements of the field before each game creating a precise 3D model of the field and its position relative to the cameras before every game [2].

With the combination of exactly where the cameras are pointed and a precise model of where the cameras are relative to the field, the ”1st & Ten” system is able to know exactly where the field is in it's frame. Then using proprietary software and mathematics the system is able to draw a line across the field [2].

The way that the NFL creates this effect, ensuring that the line is not drawn over any of the players or objects on the field uses a very similar method to this paper. It compares the color of each pixel where the line should be to the predetermined color value of the grass that should be there [3]. If the color matches the background field it will draw over that pixel with the yellow first down line and if it is some other color it will leave it as is, since it is not part of the field [2].

The way this is done for professional leagues requires extra time and money to account for the specialized equipment and needed predetermined measurements. For these reasons, this system is not ideal for a lower-budget broadcast of a football game. As such, many local high school games broadcast on local news channels do not have the first down line effect added in, resulting in a worse viewer experience. That is where this project comes in to be able to fill the need of any lower-budget football broadcast that desires having this beneficial added effect.

III. Background

The two main technologies used in this paper are the Harris Corner Detector and homographic transformations.

A. Harris Corner Detector

The Harris Corner Detector is a corner detection algorithm that works based on finding locations in an image where its x and y gradients are high. The algorithm looks all over the image breaking it down into individual small windows. In each of these windows it measures the change of intensity using the eigenvalues of the window [4]. We see that:

$$w(x, y) \text{ is the window at point } (x, y)$$

$$I(x, y) \text{ is the intensity at point } (x, y)$$

$$M = \sum_{x,y} w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

$$R = det(M) - k(trace(M))^2$$

Where R is our final score and k is a hyperparameter. We can then take this value for every point in the image and apply a non-maximal subtraction technique, ensuring we are only with the most corner-like points in local areas of the image. The remaining points can be limited by a threshold ensuring we are left with only the highest values found from the corner detector [4].

B. Homographic Transformation

A homographic transformation, or a homography, is a image manipulation technique that is able to warp pixels from an image from one shape to another [5]. This technique is used in a multitude of use cases from calibrating a camera to remove lens distortion, or in stitching two images together [6].

The concept behind a homography is that we are taking what is known as the camera coordinates of every pixel and warping them by some translation, rotation, and dilation to get the endpoint. This means we are able to warp each pixel from one plane to another plane that we can define however we would like. Our resulting image s can be represented by:

$$s \begin{bmatrix} x^{'} \\ y^{'} \\ 1 \end{bmatrix} = H \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

This transformation is used to simulate a planar surface being viewed from a different camera position as can be seen in figure 1 [6].

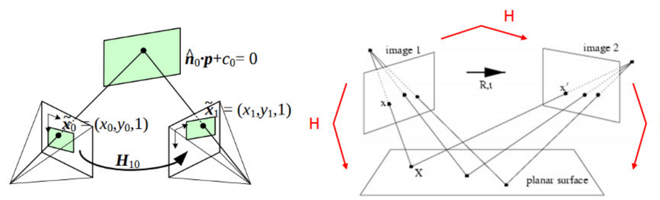

Fig. 1. Diagram showing how camera coordinates can be transformed from one plane to another using a homographic transformation.

IV. Project Design

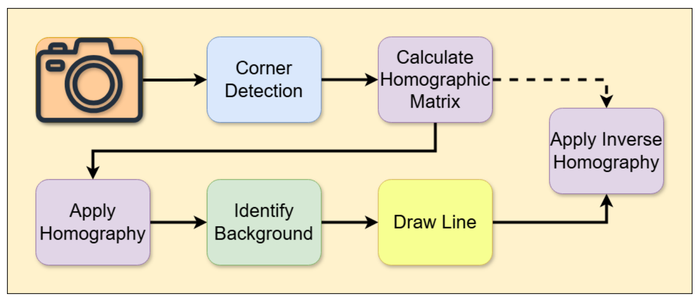

Fig. 2. Flowchart outlining the individual components of the project and how they flow into one another.

The design of this project can be seen in figure 2. This flowchart outlines the overarching structure of this project as a whole and how each individual component interacts with one another. For an actual implementation the frames of the video should be processed in parallel to ensure maximum efficiency.

A. Camera Input

The first part of the project is taking in the frame from the camera which should be pointed in the direction of the football field. There is not much too this, however for the purposes of this project the whole entire football field needs to be in the frame, specifically all four corners of the field need to be in the frame for it to work. This is not ideal as it often is preferred to focus on where the play is happening, however this is not possible for this project unless additional processing is done to zoom in on a specific part of the output image focusing on the play.

B. Corner Detection

The next step of the project is to identify the corners of the field. Preprocessing is done on the raw image to ensure we only attempt to find corners located on the field itself and no where else. We do this by only focusing on the green part of the image, denoting the grass on the field. A mask of the green pixels in the image is created to attempt to single out the field. Erosion is then performed on said mask to eliminate any stray green pixels in the background. We then dilate over a larger range than the erosion to include the whole field in the mask, even parts with obstructions. We then take the pixels from the original image from the mask and then can start the actual corner detection.

For the corner detection we use the Harris Corner Detector seen in section III-A. This results in every corner on the field being identified including all the corners between the yard lines of the field and the sidelines. Our goal is to narrow down all these corners and find the four outermost corners of the field. To do this, we find the detected corners closest in distance to each corner of the image. Since we are only looking at the field when this is done, this means we are able to find the position of each of the four corners of the field in the image.

C. Calculating and Applying Homographic Matrix

Once we have the four corners of the field we can start trying to determine where we should draw the first down line. To make the line straight across the field and parallel to the yard lines, we need to warp the field into a perfect top-down rectangular view. This can be achieved through homographic transformation, in which each pixel in the image is warped from one point to another. We can take the four corners we found and calculate the homographic matrix that will warp the field into the desirable rectangular shape. In addition to this, we can reverse the process and calculate a matrix that will warp the image back from a perfect rectangle to the original shape it resembled.

The homographic matrix can be calculated using the principles found in section III-B. Once we have this matrix, we are able to go through each pixel in the image and multiply it's camera coordinates by this matrix resulting in a new position in our output image. Figure 3 is an example of what this process looks like.

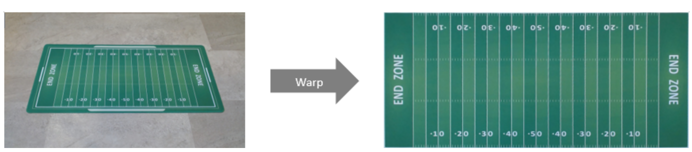

Fig. 3. Example of the result of what an image looks like after it is warped. Left side is the original image and right side is the warped result.

D. Identifying Background

At this point in the flow, we are prepared to start drawing the yellow line itself, but first we must identify the background. By identify the background, we mean identify which parts of the image are the field and what parts are not. We do this to ensure when we are drawing our line, we do not draw on top of any players or objects on the field.

The way we identify the background is by analyzing the color values of each point on the warped field. For each pixel we determine if the color of the pixel falls between two values of green, the first being the darkest shade of green on the field and the second the lightest shade of green. To make this comparison easier, we convert the standard red, green, and blue (RGB) color space to the hue, saturation, and value (HSV) color space, allowing us to focus in on the hue of the color. If the pixel value falls within this range we know it is part of the field and if it is not we know that it is something else on top of the field.

E. Drawing Line

Now having the warped perspective field and the defined background we are able to start drawing our line. We can establish that that the field is 122 yards long total. Ten yards of this for each end zone plus one extra yard for the border around the field. Additionally, 100 yards for the playing area, 98 of which are valid yard lines for a first down. We are able to use the following equation to identify what horizontal position of the image we need to start at to draw our first down line, d, given the desired yard line, y, and width of the warped image, w.

$$d = \frac{w * 12}{122} * \left(\frac{y}{100} * \frac{w * 98}{122}\right)$$

With this result d we are able to draw a straight vertical line from the very top of the warped image to the bottom of the warped image at the horizontal location from d. The result in this can be seen in figure 4

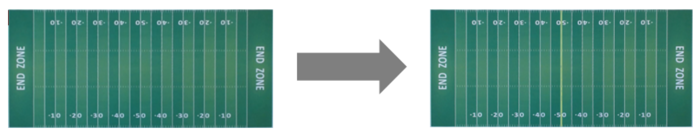

Fig. 4. Example of the result of what an image looks like after the line is added to the warped image. Left side is the original warped image and right side is the warped image with the added line.

F. Applying Inverse Homography

From this point the only thing left to do is to warp the field back to its original shape and replace the original field with our new field, yellow line and all. This is done with the inverse homographic matrix we calculated earlier and is done using the same principles that was used to warp the original image to the rectangular shape just in reverse. The small problem with this is that the new unwarped image will only have the field in its view and the rest of the background will just be black. To return the old background to the image with the line we just replace any black pixels with the background given in the original unedited frame. Figure 5 is an example of what the final result looks like.

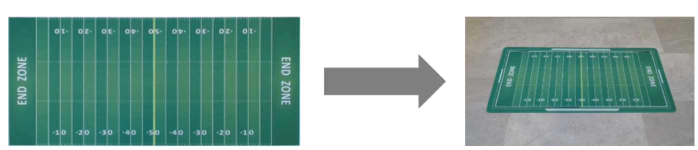

Fig. 5. Example of the what the final result looks like after. Left side is the warped image with the added line, the right side is the unwarped final result.

V. Experiment

A. Overview

For this project we were unable to get access to an American football field to collect data or test the algorithm's effectiveness. To substitute for this, a model football field seen in figure 6 was used. The model football field is a table top desk mat with a print of a football field on top of it. The field is 35.4 by 15.7 inches and is proportional to a real football field's dimensions. This example field will be used to create a dataset of several images and to test how the project works in a real-life scenario.

Fig. 6. Image of the model football field being used for the experiment.

B. Setup

The setup of the experiment is designed to replicate a real-world scenario as close as possible, despite using a model field much smaller than an actual field. While this project is designed to work in many different configurations, the most practical setup will involve placing the camera one or two feet away from the field on a tripod that raises the camera about two feet in the air. The camera should be tilted slightly down such that the entire field is in view of the resulting image. For testing, a Logitech C920x webcam was used for creating a dataset and all video testing done.

C. Dataset

Since the accuracy of post-processing effects in video can be very hard to quantify, our experiment will be tested on a dataset of images, with supplementary metrics based on the real-time video processing performance. The dataset consists of 200 images total, 100 bare images of the field, and 100 images of the field with some sort of object on the field to test how well the project works with occlusions. Each image in the dataset is differentiated from one another, each being taken at different camera angles, different positions of the camera, and different lighting conditions. The angles are varied in both pan and tilt angles. The position of the camera is varied by distance from the field, height above the field, and horizontal position relative to the field. The lighting conditions are varied by differing levels of artificial and natural lighting.

Each image in the dataset will be processed individually with the output image for each input image being recorded. These output images will then be human evaluated to either contain a valid line or an invalid line. An image is defined to contain a valid line if the image of the field contains a yellow first down line placed on the field in the correct position, parallel to the yard lines on the field, and not being drawn on top of any occluding objects that may be on the field.

D. Additional Metrics

In addition to the metrics evaluated from the dataset, metrics based on the performance of the real-time video processing will be used to help analyze the efficiency of the algorithm. These metrics will include the average frame rate and average delay induced from processing each frame. All the testing for these metrics will be performed on a computer containing an Intel Core i7 10th generation processor and 16 gigabytes of ram. As is expected, running the algorithm on a computer with better hardware will result in a higher frame rate and less delay while the opposite is true for a computer with worse hardware.

VI. Results

The resulting metrics found from the testing done on the dataset created for this experiment are as follows. 88% of the images of the normal field were determined to have a valid line and 83% of the images of the occluded field were determined to have a valid line.

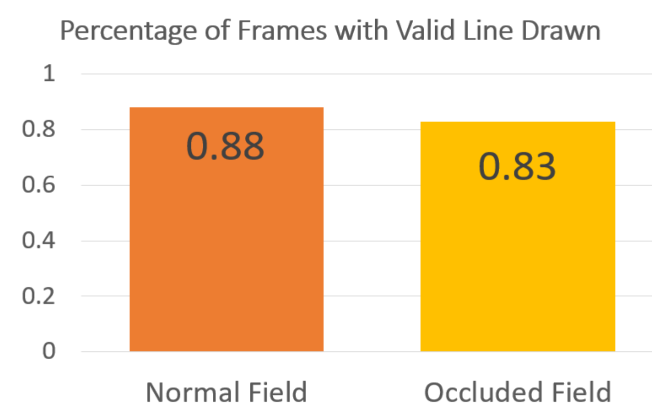

Fig. 7. Graph showing the results from the experimental test on the dataset discussed in section V.

The additional metrics were determined and the results can be seen in table I.

TABLE I
TABLE SHOWING THE RESULTS OF THE ADDITIONAL METRICS, THOSE BEING THE AVERAGE FRAME RATE AND THE AVERAGE DELAY EACH FRAME.

	Frames Per Second
Average Frame Rate	20.23

	Seconds
Average Delay	0.347

VII. Discussion

The results seen from the analysis of the dataset are reasonably good. We see that on a field with no objects obstructing its view we get a valid line being drawn 88% of the time, while on a field with occlusions we get a valid line only 83% of the time. It is expected that the occluded field would have a lower accuracy than the normal field as it only has more challenges to face. The need to draw the line in the exact same way as the normal field but also ensure it is able to not draw the line over top any occluding objects, strictly means there is more complexity for the occluded field.

To provide further analysis of the dataset it is the case that a majority of the images that did not result in a valid line being drawn had extreme conditions that caused this to happen. It was often a very extreme camera angle or sharp changes in lighting due to shadows being cast on the field that cause the first down line to either not be successfully drawn or not drawn in the correct place. We believe the reason behind this is that these conditions cause the Harris Corner Detector to not be able to pick up on the corners of the field. Sharp camera angles, such as angles too low to the ground or angled too much relative to the field, cause the corners to be less recognizable to the Harris Corner Detector algorithm. Similarly, stark shadows on the field cause light spots and dark spots on different parts of the field making it hard to identify which shade of green is representative of the field. This means if the shadow is near a corner sometimes that area of the field can be removed from the preprocessing done before the corner detection, making it impossible to identify that corner. Additionally, in some cases where the field had two different brightnesses due to shadows, the background identification would sometimes see one of the sections as being an object and not part of the background, therefore not drawing the first down line in that section. In a practical use setting as long as the user knows these limitations, it is easy to avoid them. The camera can be placed in a high enough location, looking perpendicular to the field, with minimal shadows for optimal results, as every single image in this orientation successfully drew a valid line.

The additional metrics of average frame rate and average delay are potentially less desirable than the accuracy of the model. The camera being used for our testing recorded at 30 frames per second so getting output video of only around 20 frames per second is not an ideal result. Many optimizations were made in our implementation to attempt to increase the frame rate and reduce the delay, and while they did improve these parameters they are still not perfect. While this project was designed to work with any reasonable hardware, it seems to run at full efficiency this project may need higher end hardware. That being said, the need for better hardware is still much less costly when compared to the cost requirements of the NFL's "1st & Ten" system [1].

VIII. Future Work

Not everything about this project is perfect. With some further work, there could be some additions that make the model much better. The first thing that would improve the model much more would be adding functionality for identifying where the camera is pointed on the field without needing all four corners of the field. This could be done by doing number recognition on the section of the field you are looking at to get an understanding of where you are on the field. Corner detection could then be used in a similar way to how the project works now to precisely place the first down line to be exactly where it needs to be.

In addition to this further testing on actual real football fields could be used to help tune the model's implementation and ensure that it is working correctly for the use case it is designed for. This extra testing would allow for additional confirmation that the model is working as intended, or prove that it could use some adjustments to make it better.

IX. Conclusion

As seen from the project analysis and results found from testing, this paper outlines a very suitable alternative way to add the virtual first down line to an American football broadcast. This project processes video footage in real-time providing output with the desired result in a significant percent of the time. While there are some scenarios that result in inaccuracies in the model, these scenarios can be avoided by a well-informed user who knows some of the limitations of the model.

For these reasons this system is able to be used in real-life scenarios, that being lower-budget football broadcasts, to improve the broadcast and improve the viewer's experience. This is a product that fills a niche that is currently not explored anywhere currently in an effective way. This therefore could be useful to a larger number of users, specifically small local television stations or small internet broadcasters that would like to increase the production value of their broadcast with a simple effect.

Acknowledgment

I would like to thank all the professors at RIT that taught and supported me through my college career. I specifically would like to thank Dr. Thomas Kinsman for being my academic advisor for this project, Dr. Minseok Kwon for helping me throughout the capstone process, and lastly Dr. Hans-Peter Bischof for the great graduate program he runs. I would also like to thank my mom and dad for supporting me throughout my whole college journey.

References

[1] Wikipedia contributors, "1st & ten (graphics system) — Wikipedia, the free encyclopedia," https://en.wikipedia.org/w/index.php?title=1st_%26_Ten_(graphics_system)&oldid=1260172278, 2024, [Online; accessed 21-February-2025].

[2] J. Fong, E. Caswell, and G. Barton, "The nfl's virtual first-down line, explained," Feb 2016. [Online]. Available: https://www.vox.com/2016/2/6/10919538/nfl-yellow-first-down-line-espn

[3] B. Nagarajan and P. Balasubramanie, "An invariant approach to object classification using background removal," in *2008 International Conference on Computing, Communication and Networking*, 2008, pp. 1–6.

[4] O. Contributors, "Harris corner detector." [Online]. Available: https://docs.opencv.org/3.4/d4/d7d/tutorial_harris_detector.html

[5] D. Zhihua, Z. Zhijiang, and Z. Dan, "Indication position estimation by the homography of planes," in *2010 2nd International Conference on Signal Processing Systems*, vol. 3, 2010, pp. V3–66–V3–69.

[6] O. Contributors, "Basic concepts of the homography explained with code." [Online]. Available: https://docs.opencv.org/4.x/d9/dab/tutorial_homography.html