Translation of NFL videos into 2D perspective animation

Sibi Shanmugaraj
CS 231A Stanford University
sibiyes@stanford.edu

Abstract

In this project we undertake the task of translating NFL game play videos into 2D perspective animation in the bird's eye view that is suited for whiteboard analysis. The approach consists of detecting features and entities from the images that help us establish point correspondences, detect player location on the field of play, and compute homographies to perform translation to the perspective view that is desired. The translated view shows the player location in the corresponding perspective. And finally we wish to apply the process to videos to be able to generate dynamic animation. The methodologies involved in detecting the image features and computing the homographies is presented. The learning approaches that are used are presented as well. The shortcomings associated with the current methodologies is discussed and possible ways to overcome those and improve the accuracy of detection is presented.

1. Introduction

For CS 231A term project, I have decided to work on using video clips of NFL plays and translate them into equivalent 2 dimensional animation in the birds-eye perspective view. Football is a very popular sport in the US and is being played at the school, college and professional level and has a large following. The game involves significant level of strategic planning and analyzing your game play and that of your opponents has become a significant part of the game today.

Professional NFL teams make significant investments in video analysis to be able to analyze the game. So the application of translation of video into format that is suitable for studying is of great value. Current NFL teams have sensors to track players and be able to generate insights that they need to strategically prepare themselves. However the interest and the value of these insights extend much beyond the professional NFL teams. They are equally valuable to high school and college teams as well. But they will not be able to make significant investments to be able to place sensors or capture the game play using large number of cameras. Also with the increase in scale of sports betting, these insights will be valuable to an avid enthusiast in sports betting as well. The idea of using granular game data in making betting decisions is not new. Adding the insights and information generated from the actual game play videos will definitely add new dimensions to that. For example, one might be able to look at the running gaps that are generated by the offensive line by analyzing the videos, and this information could be valuable in understanding the game better and making various betting decisions. We can objectify expert analysis and remove bias through this process. So the idea of being able to generate 2D perspective animation of the game from limited views (1 or 2) of the game video has a great value.

1.1. Related Work:

Sports analytics and in particular video-driven sports analytics is something that is being actively used by professional teams in various competitive sports as well as by media outlets. For instance, for NFL we have a lot of analysis provided by NextGen Stats (https://nextgenstats.nfl.com/). The area has also attracted interest from academic publications as well. In [1] and [2] we can see the discussion of various approaches that go into the process of sports video analysis such as entity detection, feature detection and tracking, pose detection etc… Stephan Janssen in [3] has a similar work performed for basketball and has detailed the application of various computer vision and machine learning methods that are applicable in this context. Camera calibration can play an important role in tasks in this area. We have some related work around camera calibration in the context of sports video analysis in [4] and [5]. Detecting and tracking players is an important element in sports video analysis. Lu et.al[6] describe some approaching that can be applied to detect and track players in videos. We can very much use generic well-proven methods towards detection of players and apply them in this area and tune them to our need. The YOLO[7] model is a widely used in detecting humans in images and is used in this work as well. Sometimes we might want to detect not just the location of the players in the frame, but also their pose. The AlphaPose[8] provides us with a good framework for taking up pose detection tasks.

1.2. Problem Statement

For this project, I plan on using the game play videos available from a kaggle contest (https://www.kaggle.com/c/nfl-impact-detection/). We have 120 videos for 60 plays, with 2 views for each play. One captured from the sideline and the other captured from the endzone. Sample image frames for each view is provided in figures 1. and 2 respectively. The videos are shot at approximately 60 frames per second and has frames from both view synchronized. The dataset also has the location of the players in the frames of the video in the field of play coordinate system that were collected using sensors. These values will be used to validate the results generated from this work. A simple mean distance metric can be used in quantifying the accuracy of the methodology. Collection of further game play videos can be used for the extension of the current work. For the initial work presented thus far, the processing is done on the images of the video frames extracted from the video clips. The final version of this work will incorporate a pipeline that generates output for the entire video clip. The proposed output will have the translation of the frame to a top view of the field of play that would show the play captured by the two input views (sideline and endzone). This view resembles white board depiction of the plays that is traditionally used for years and help in easy understanding of the play.

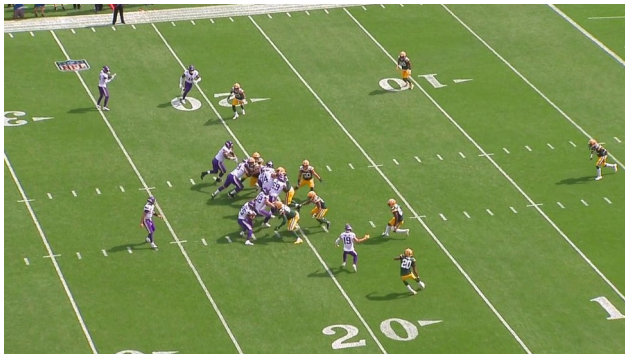

Figure 1. Image frame from the sideline view

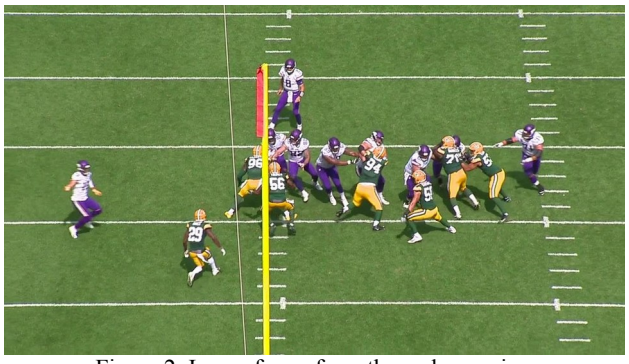

Figure 2. Image frame from the endzone view

2.0 Technical Approach

There are 3 main phases to the proposed work
(i) *Entity and feature identification:* This step involves identifying key elements from the image frames in the video such as field of play, yard lines, hash marks and players on the field. These elements enable computing point correspondences between the input and the translated output image frames.
(ii) *Computation of translation to the proposed view which is a top view of the field of play:* We use the point correspondence to compute homographies to perform the required translation
(iii) *Application of the first 2 steps to video to generate an animation of the play:* Apply the methodology to video and incorporate methods that are suited specifically for processing videos directly.

2.1 Entity and feature identification:

To perform the translation of the image of the game play to the proposed perspective, we need to establish point correspondence between the images we have and the final proposed view. Towards that end we exploit our prior knowledge of the dimensions of the football field and presence of entities or features that will help us assign relative position to points in the image. The key features towards this end are the yardlines, boundary lines and the hash marks on the field. We have yardlines marking every fifth yard and hash marks for every yard. Along the width of the field the hash marks are placed at a distance of 69.9 and 90.5 feet respectively from the sideline boundary. Identification of the these features will allow us to assign relative coordinates for points on the image in both the world reference frame and the image reference frame in the translated image. We can choose a corner of the field as the origin and assign relative coordinates to the points identified. In this case the origin is chosen to be the top left corner of the field with respect to the sideline view.

Identification of yardlines is achieved by making use of the fact that the yardlines are straight lines and we can apply line detection methods after applying some pre-processing methods. In this approach the image is converted to grayscale and canny edge detection filter is applied first to detect edges that would define a line. The yardlines are white in color and have a contrast with the field of play which is green and they form clear edges that could be detected by the canny edge filter. Hough transform is applied to the edges and a suitable thresholding is applied to be able to detect peaks that would correspond to the yardlines. However there are other noises in the image, especially when the image has regions outside the field of play as well. The hough transform identifies line features outside the field of play as well. To remove them we apply filtering on the angle parameter in the hough space as the yardlines tend to fall within a range of angles. There are also other line features which end up getting detected in the field of play as well

that aren't filtered out. This issue needs to be addressed as well to make the detection even more accurate. Another issue is that a yardline might be detected as 2 lines due to the fact that we are operating on edges. Some simple heuristics is applied to filter these cases by keeping one of the line and removing the other (Lines that begin and end close to each other in the range of the image). Color based filtering on the HSV scale to identify lines didn't work well enough as there are a more variation on the field due to different arenas, wear and tear of the field, lighting, shadow etc.. Figure.3 shows detected yardlines.

The boundary lines are detected by applying the appropriate angle filtering and some selected heuristics. We assume that the upper boundary line begins or ends within the top 30 percent of the image frame and the lower boundary line is in the bottom 30 percent of the frame for the sideview. The lines that clear the angle filtering and closest to these demarcations are identified as boundary lines. This method is more prone to noise and needs further improvement. Identification of boundary lines serves 2 purposes. First as a feature of interest that in combination with the yardlines will allow us to assign relative positions in the translated image frame by computing their points of intersection. Second they help us identify the field of play and hence allow us to filter out region outside the field of play that help us eliminate noise for other identification steps later.

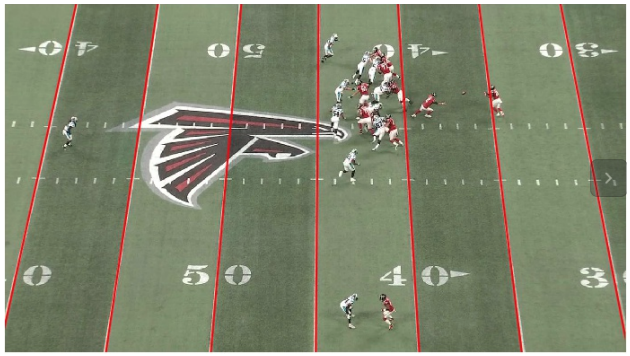

Figure 3. Detected yardlines in sideline view

The next key feature that we need to identify are the hash marks. The line through hash marks are perpendicular to the yardlines in the 3D world and the output perspective. The intersection of the line through hash marks and the yardlines can be assigned coordinates with respect to the selected origin. To fit a line through the hash marks we first need to identify points that correspond these hash marks. We first filter out the image to exclude the regions that are outside the field of play based on the boundary lines detected previously. We then apply gradients on the image and filter out points corresponding to a specific region by choosing specific gradient ranges to generate a mask, that identifies larger and more prominent entities in the field of play such as the players and the logo. A binary erosion and dilation is applied to make these gradient based masks more prominent and an improved mask is generated as a result of this. This mask removes the entities that corresponding to the larger gradient while keeping hash marks and some other entities. We then apply another round of gradients , this time incorporating the mask generated from the previous step and choose a different gradient range and apply some dilation to identify the hash marks (as well as other distinct patterns). This will create distinct blobs for the hash marks. But the blobs are created for other elements such as yard markings and noisy regions due to color and texture variations in the field of play. An example of the resulting filtered binary image is shown in figure 4. Now we apply difference of gaussian blob detection method to identify blobs. Figure.5 shows the blobs detected using this approach. The laplacian of gaussian and determinant of gaussian methods for blob detection were also tried but the difference of gaussian worked best in this context. A simple heuristic based filtering is used to pick the blobs that correspond to hash marks. The filtering is not completely fool proof, but removes very obvious blobs that are not hash marks. The points corresponding to the detected blob centers are used in another round of hough transform fitting. A suitable thresholding, angle filtering in the hough space will allow us to fit lines that identify the hash marks. The fitted line will pass through the hash mark blobs. It has to said that, though this method shows some success in identifying the hash marks, there are instances were the method fails and instances where it fits noise that doesn't correspond to the hash marks. Figure 6. shows all the detected lines that are of interest based on the methods discussed thus far.

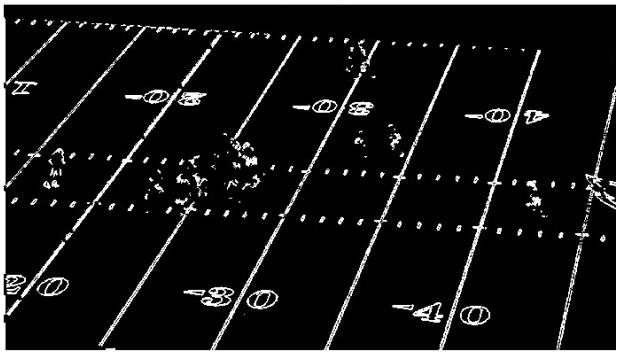

Figure 4. Applying filtering using gradients, erosion and dilation to highlight blobs of interest (hashmarks)

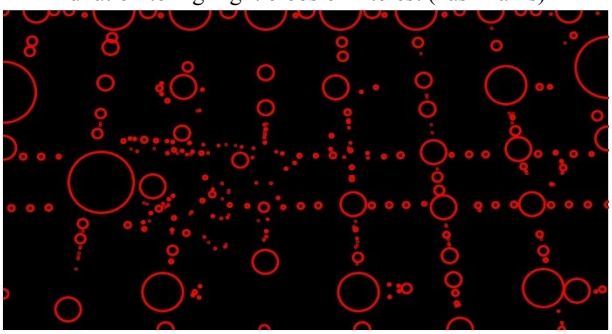

Figure 5. Blob detection output using difference of gaussian method

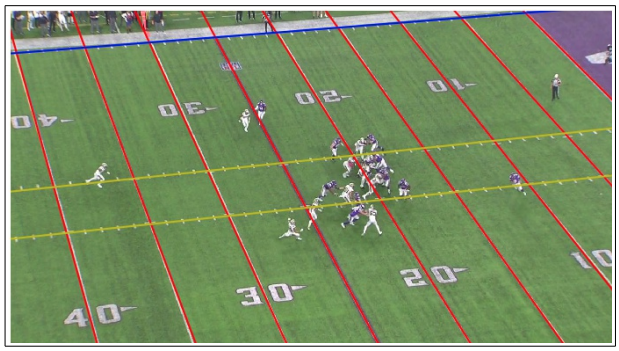

Figure 6. All the detected lines. Yardlines in red, hashmark lines in yellow and boundary line in blue

Once we have identified the yardlines, we also need to associate the identified lines to the specific yard value that corresponds to the line. For this purpose, the yard marking on the field that correspond to the line can be made use of. The first method adopted to detect the yard markings is to apply pattern matching using normalized correlation. A sample of sub images that correspond to the yard markings are extracted for all possible yard markings (in this case 10, 20, 30, 40 and 50). We use these sub images to pattern match with the image to identify the yard markings. The method is not accurate enough when it comes to detecting the regions that correspond to the specific yard marking, identifying the exact yard mark. However it does a good enough job in identifying image regions that correspond to yard markings. In order to increase the likelihood of detecting the all regions corresponding to the yard value, we select a sample of 5 sub images of the yard marking such that we have a sub image for each possible yard value. We apply template matching on the image using each of the sub images and then perform a peak selection to select points that have high correlation with the sub images chosen. These points and the bounding box corresponding to them are estimated as the region corresponding to the yard marking. Since we perform pattern match using multiple images, we may get matching points that are close (from matching with multiple sub images) to each other that correspond to the same yard marking. To resolve this, we consolidate peak points that are within a specified distance of each other by their mean value. Some of issues that we face with this task are due to the varying orientation and the relative size of the yard mark region that vary with the zoom level and the angle of focus of the image frame.

Once we identify the regions that correspond to the yard marking, we need to identify the yard value that they represent. So to address that, a learning approach is being adopted where we build a training model to identify the correct yard value or if the region is noise, on the identified image region extracted using the method described above. We achieve this by building a conv-net model to perform the classification. The model is pre-trained on MNIST dataset and then trained on the dataset associated with this task. We use a training dataset with 20 images for each class and perform some rotations for data augmentation. The current version of the model is not 100 percent accurate. We achieve a training accuracy close to 90% and validation accuracy close to 70%. We can address the accuracy issue by training on larger dataset and by fine tuning the model design and parameters. Figure.7 shows the correlation heat map after applying pattern match with the sub image. Figure 8. shows the result of applying peak selection to identify the region that correspond to the yard marks.

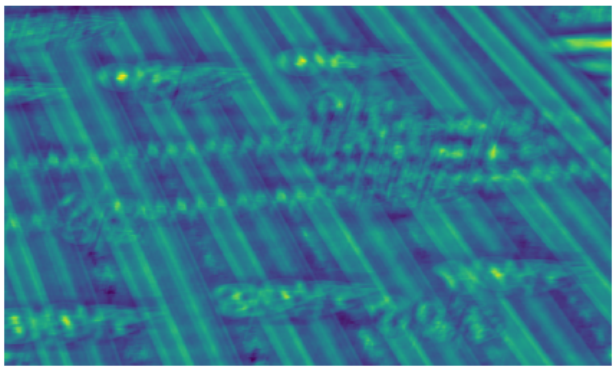

Figure 7. Correlation heatmap after applying pattern match

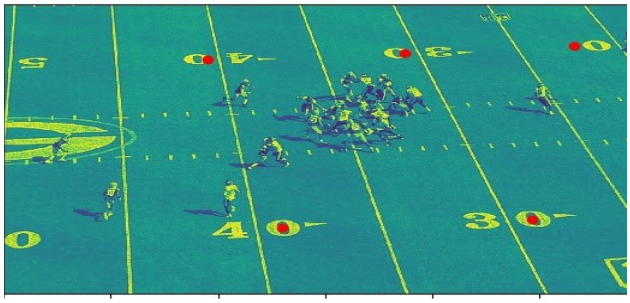

Figure 8. Yard mark regions estimated after applying peak selection on the correlation output

The next key entity that needs to be identified is the actual players on the field. For this purpose we use a pre-trained YOLO model that can identify humans among other class entities in an image. The output of YOLO is in the form of bounding boxes defined by [left_x, top_y, width, height]. The output from prediction using YOLO model is shown in figure. 9. From the bounding boxes computed, the location of the player is estimated to be the midpoint of the bottom line.

This way of estimating the location of the player does come with its own error and results in an ambiguity between both the views we have. This is because the person in a 3D world should be bounded by a cuboid but in a 2D image, we have a rectangular box bounding the person. If we take the length of the field to be the u-axis and the width of the field to be the v-axis, the estimation of the location from the bounding boxes in the sideview gives a much accurate estimate of the position in the u-axis but has an error along the v-axis. This is because the bounding box in this case corresponds to the side of the bounding cuboid that is closest to the side view perspective.

Similarly, the estimated location from the bounding boxes in the endzone view gives a more accurate estimate of the position along the v-axis but has an error along the v-axis. If we combine the estimates from both the views and pick the u-coordinate from the sideview and the v-coordinate from the endzone view we would get a much better estimate of the player location on the field of play and resolve the ambiguity (to an extent) associated with a specific view. Towards that end we also need to identify the identity of the players to match them between the views. This scope is not addressed in the current work, but an appropriate learning-based approach with properly tagged data is a good candidate to begin with.

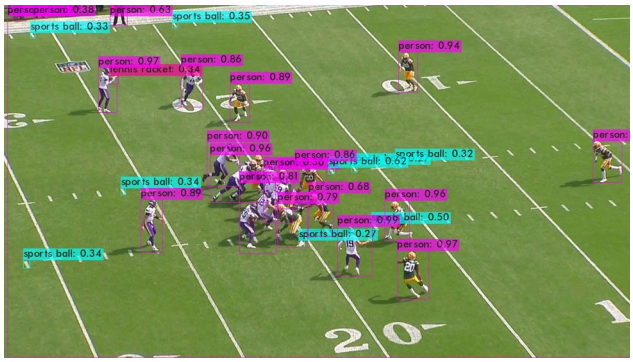

Figure 9. Bounding box outputs from YOLO

2.1.1 Issues with entity and feature detections:

The above methods that are used to detect features such as yardlines, hashmarks, players etc… have issues of their own.

For starters, the line detection that is critical in detecting the field of play and yardlines has noise associated with the detection. There are instances were the lines of interest have gone undetected and instances were undesired lines get detected. The blob detection methods applied in detecting hash marks is also not accurate and is noisy. We fail to detect blobs of interest. We also end up detecting blobs that are not hashmarks as there are various patterns that manifest themselves as blobs. We can address this issue by employing learning based methods to detect the field of play, yardlines and hashmarks. Semantic segmentation and bounding box models can be trained to address these tasks.

The predictive model that we use to predict the yard value of the yardmark regions is not 100% accurate. This model is built on a training dataset that has 20 images for each of the classes (5 possible yard values and outlier regions that are not yardmarks) while applying some rotations for data augmentation. We can improve the accuracy of this predictive model by training on more data. Another aspect that we need to incorporate in this modeling approach is to identify the direction of the yard value (for example, 40 yards to endzone in a specific direction). This can be detected by identifying the arrow mark next to the yard value.

The YOLO model that is used to detect the players in the field fails to detect all the players when they are crowded together. This results in us not being able to detect all the players. We need to address this problem and one way to to do that is to retrain the YOLO model for this specific use case where we tag the data in such a way that all the players are independently tagged. One option to do that is to tag the helmets of the players and retrain the YOLO model to predict bounding boxes for helmets as well. We can also explore optical flow methods to track players across frames so that if a player is detected in one frame we can translate that across frames. We can also explore other low-level feature based detections to detect players in a more robust manner.

In order to compensate for these limitations we manually tag the yard value corresponding to the yardlines to generate point correspondences and use them in the subsequent homography computation.

2.2 Computation of the translation homography:

With the point correspondences generated, we can compute homography for translation from the view in the video frames to the perspective top-view view defined. A simple least sqaures approach is adopted to compute the homography H.

The homography is computed from the point correspondences as shown below.

Let the points in the input images be $[(x_1, y_1), (x_2, y_2), \ldots.(x_n, y_n)]$ and the points in the translated images be $[(x_1', y_1'), (x_2', y_2'), \ldots.(x_n', y_n')]$. The computation of homography based on these point correspondences is shown below.

Least squares methods

- fitting an homography -

$$h_{11}x + h_{12}y + h_{13} - h_{31}xx' - h_{32}yx' - x' = 0$$
$$h_{21}x + h_{22}y + h_{23} - h_{31}xy' - h_{32}yy' - y' = 0$$

From n>=4 corresponding points: $Ah = 0$

$$\begin{pmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1x_1' & -y_1x_1' & -x_1' \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1y_1' & -y_1y_1' & -y_1' \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x_2x_2' & -y_2x_2' & -x_2' \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -x_2y_2' & -y_2y_2' & -y_2' \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n & y_n & 1 & 0 & 0 & 0 & -x_nx_n' & -y_nx_n' & -x_n' \\ 0 & 0 & 0 & x_n & y_n & 1 & -x_ny_n' & -y_ny_n' & -y_n' \end{pmatrix} \begin{bmatrix} h_{1,1} \\ h_{1,2} \\ \vdots \\ h_{3,3} \end{bmatrix} = 0$$

We use the point correspondences to compute the homography. Once the homography is computed, we apply the translation to the player location points using the computed homography. This allows us to place the player location points on the translated bird's eye view. The estimation of location of the players on the field is the

most critical scope of this work and being able to compute the appropriate translation from the input images that are from the videos to the translated view that is proposed enables us to do that estimation. The obtained translations for the corresponding frames from the sideline view and the endzone view are shown in figure.10 and figure.11 respectively. We can see a difference in the number of players detected between each views. This is due to 2 reasons. First being that not all players are visible in both views especially the endzone view being much more focused than the sideline view. This is evident from the observation that the translation of the endzone view has lesser number of players. Another reason for this disparity is the issues associated with the detection of players using the YOLO model. As mentioned before, not all players are uniquely detected especially when they are crowded together. This would lead to some players being detected in one view and not being detected in the other. Given that we just need one final translated view and the benefit of having two views is to be able to resolve the ambiguity associated with the 2D projection in a specific view, we can use the information available from both views to resolve the ambiguity as much as possible. So we would still have some errors in the player locations when we have a player not captured in both the views. This is a downside associated with this methodology as we can't know the location of the player if they are outside the view of the frame.

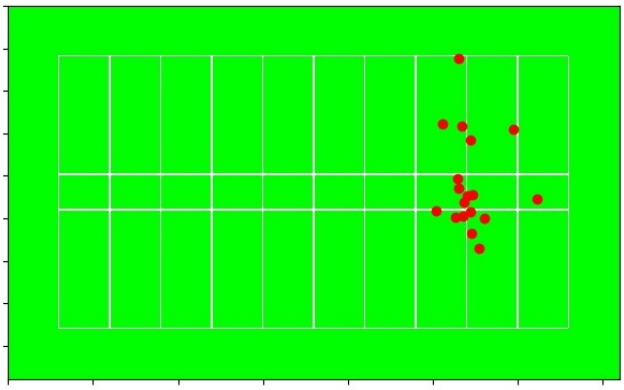

Figure 10. Translation obtained from the sideline view

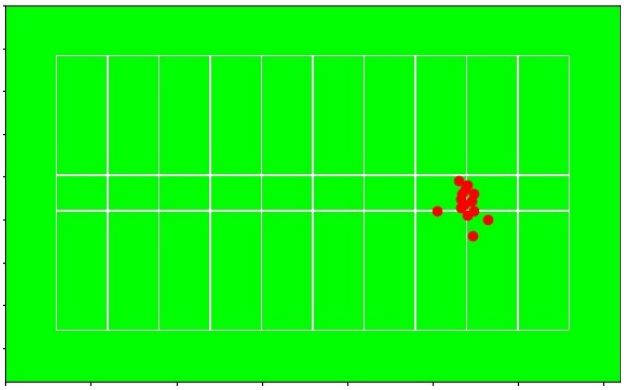

Figure 11. Translation obtained from the endzone view

2.3. Application to video:

The extension of this work is to apply these methods to the video as a whole and produce a dynamic animation showing the play in the bird's eye view. Though we can statically apply these methods to each image frame in the video, we can also exploit optical flow methods to track entities and features across frames to improve the quality and accuracy of the detections. This could help address some of the limitations we have discussed. We can also explore sequence models and LSTMs in the context of learning-based approaches that are required and proposed in this work.

2.4 Results:

The results presented in this work are qualitative in nature due to the nature of the task and we are able to see the detection of the features and entities along with the intermediate outputs through the images provided. We do have the sensor captured location of the players in the video frames available with the dataset. However due to the limitations discussed, we are not able to identify all the players uniquely along with their identity. Once we are able to identify all the players uniquely in terms of both their identity and location we can use the known locations to compute error metrics such as mean-squared error to quantify the accuracy of the approach.

3.1 Conclusion

In this work we showed how we can make use of limited views of the NFL game play to be able to generate whiteboard birds-eye view which is widely adopted in game analysis and strategy preparation. Since not everyone who would find interest and value in such information can have access to multiple camera views or high quality sensors, especially when it comes to an average spectator who doesn't have any ability to control the number of cameras or the sensors used or get access to them, the possibility of a simpler approach even if it has some limitations is encouraging.

In any image translation problem the most important element is to be able to identify point correspondences. That was shown to be the case in this work as well, as we were able to perform the translation necessary once we were able to compute those correspondences. Another important issue is the possible ambiguity that arises with a 2D projection and how having multiple views allows us to resolve the ambiguity as much as possible. We saw the ambiguity that we faced with a single view and how having two views allowed us to resolve the ambiguity to an extent. As a natural extension, if we have additional views we may be able to make the process more accurate and robust.

Even though this work is primarily an image translation work based on point correspondences, the task associated with computing the correspondences is a computer vision problem ranging from feature and entity detection, filtering and other image processing tasks to learning-based methods. We discussed how we can use line detection,

blob detection, gradients and various other methods to detect entities of interest. We discussed the limitations associated with those methods and the possibility of employing learning-based methods such as semantic segmentation, bounding box detection and other similar ones to address those problems and come up with more robust and accurate detection.

We also saw how we may need to use learning based methods such as YOLO to detect person and yard marks in the process. The context may vary across different sports but we still need to detect the players and if needed their pose as well. So the application of these methods finds a place in these kinds of tasks. Overall we are able to see that, the task of identifying the point correspondences is a combination of multiple tasks that are applied together. This shows that any practical image translation problem is more likely to extend into a larger scale computer vision problem with application of various computer vision and learning methods.

Finally we discussed the scope associated with applying these methods to a video input. Though many of the discussed methods can be extended in some way to video inputs, we can also explore methods that are much more suited to video inputs directly. The possibility of using sequence models such as LSTMs, optical flow methods and other similar methods could be explored.

3.2 Project code repo:

https://github.com/sibiyes/cs231a_project

4.1 References:

[1] Huang-Chia Shih, A Survey on Content-aware Video Analysis for Sports, IEEE Transactions on circuits and systems for video technology vol. 99, no.9, January 2017, *https://arxiv.org/pdf/1703.01170.pdf*

[2] Colby T. Jeffries, Sports Analytics With Computer Vision, https://openworks.wooster.edu/cgi/viewcontent.cgi?article=10456&context=independentstudy

[3] Stephan Janssen, Open Source Sports Video Analysis using Maching Learning , *https://dev.to/stephan007/open-source-sports-video-analysis-using-maching-learning-2ag4*

[4] Pei-Chih Wen, Wei-Chih Cheng, Yu-Shuen Wang et.al, Court Reconstruction for Camera Calibration in Broadcast Basketball Videos, Journal of Class Files, Vol. X, No. X, January XXXX, *https://people.cs.nctu.edu.tw/~yushuen/data/BasketballVideo15.pdf*

[5] Jianhui Chen, Fangrui Zhu, James J. Little, A Two-point Method for PTZ Camera Calibration in Sports, *https://www.groundai.com/project/a-two-point-method-for-ptz-camera-calibration-in-sports/1*

[6] Wei-Lwun Lu, Jo-Anne Ting, James J. Little, Kevin P. Murphy, Learning to Track and Identify Players from Broadcast Sports Videos, IEEE Transactions on Pattern Analysis and Machine Intelligence, *https://www.cs.ubc.ca/~murphyk/Papers/weilwun-pami12.pdf*

[7] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, You Only Look Once: Unified, Real-Time Object Detection, https://arxiv.org/pdf/1506.02640.pdf

[8] Shanghai Jiao Tong University, Machine Vision and Intelligence Group(MVIG), AlphaPose, https://www.mvig.org/research/alphapose.html.