Reproducing Raw NBA Tracking Data Through Computer Vision

Nehal Chigurupati

December 2023

Contents

1 Objective

This project was motivated by the desire to work with raw tracking data past 2016, the last year the point-velocity format was made publicly available. The objective is threefold:

- Produce a model that can identify and track people
- Generate coordinates and velocities for players
- Use the coordinates and velocities to classify play types (both simple, like pick and roll, and complex, like motion strong)

2 Model

YOLOv8 was used to identify players, while DeepSORT was then used for tracking. Several modifications had to be made to DeepSORT for it to function properly with newer versions of Scikit-learn and TensorFlow:

- Change variable types from np.float, np.int, etc. to built-in float, int types.
- The Scikit-learn linear assignment method has been deprecated, requiring the Munkres algorithm to be hard-coded directly in the DeepSORT implementation.
- Adjust tf.Session to reference TensorFlow v1. Sessions were deprecated in v2.

3 Reproducing position and velocity

3.1 Position

Position for a given player was given by a point (x, y) in the plane, calculated to be the center of YOLOv8's bounding box. Because the videos of the court are captured at an angle, distances tend to dilate linearly towards the side of the court closer to the bottom of the frame. To solve this problem, more complicated techniques involving homography are needed (work in progress). For now, however, the coordinates of the players in the image are sufficient to work with.

3.2 Velocity

Velocity in the x-direction was calculated by subtracting the last known x-position from the current x-position and dividing by the number of elapsed frames since the last known x-position was recorded. Velocity in the y-direction was calculated analogously.

4 Play classification

4.1 Hard-coded approach

4.1.1 Screens

Informally, to identify whether a screen has been set at time t by player p_1 for p_2, the tracking data from a sufficiently small time span around t is searched for the following characteristics.

- p_1 has velocity near zero while p_2 has non-zero velocity and is moving towards p_1. If this situation occurs, followed by p_1 and p_2 getting "very close together", then a screen has been set.

- p_2 has velocity near zero while p_1 has non-zero velocity and is moving towards p_1. If this situation occurs, followed by p_1 and p_2 getting "very close together", then a screen has been set.
- p_1 and p_2 both have non-zero velocity and are moving towards one another. If this situation occurs, followed by a period of time where either p_1 or p_2 is stationary, then a screen has been set.

To make this algorithm more rigorous, it's necessary to more precisely determine a few terms.

First, we need to decide what a "sufficiently small time span around t" is. This interval $[t_0, t_1]$ is meant to be the longest length of time we expect the setup for a single screen to take. As a generous first attempt, we can try five seconds, and refine from there.

Second, we need to decide what a "velocity near zero" means.

4.2 Convolutional Neural Networks (CNNs) approach

5 Application: Spacing

For a video demonstration, see:

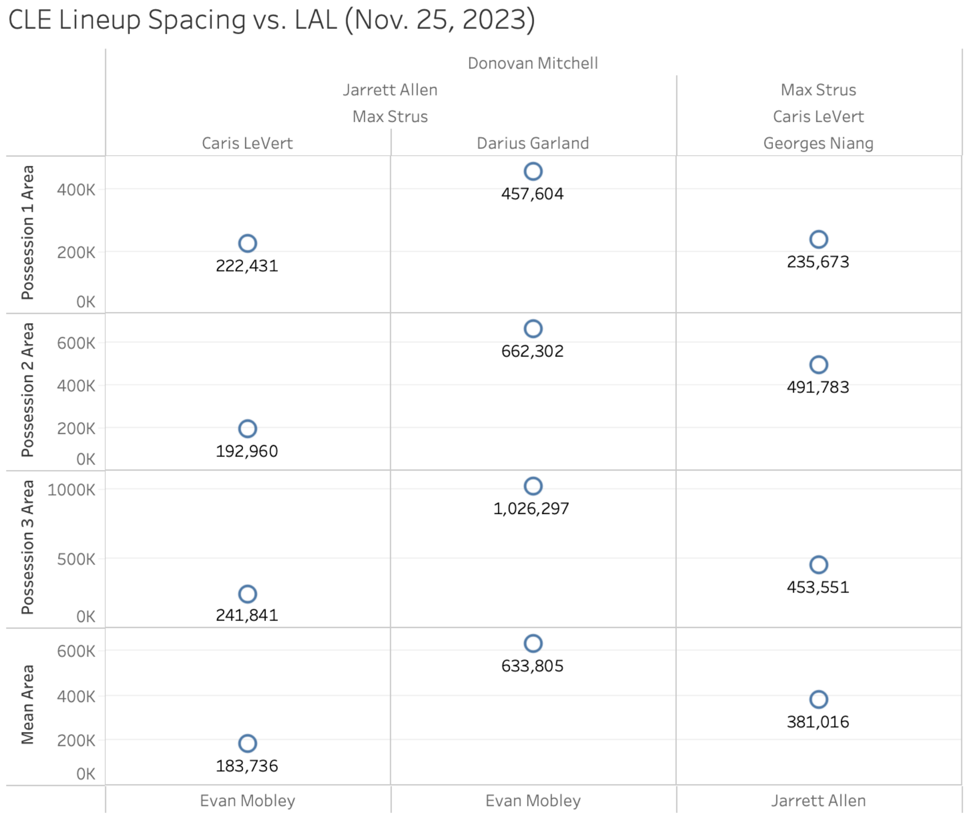
https://drive.google.com/drive/u/2/folders/1q2OZWOXFuLCJOT9Ykjw3q1wml40U2hw0

For a given possession, the defenders were identified and associated with the vertex at the center of their box. The area of the convex hull spanned by those points then served as an approximation of area of the driving lane - i.e. how much "spacing" there was on the floor.

To see how effective this metric was (referred to as convex spacing), three Cleveland lineups from the November 25th, 2023 LAL @ CLE matchup were tested. This game was selected because of the relative good health and availability of the Cavs squad.

- Donovan Mitchell, Jarrett Allen, Max Strus, Darius Garland, Evan Mobley
- Donovan Mitchell, Jarrett Allen, Max Strus, Caris LeVert, Evan Mobley
- Donovan Mitchell, Max Strus, Caris LeVert, Georges Niang, Jarrett Allen

For each lineup, four different offensive possessions were located. All of the possessions involved half-court sets (i.e. no transition plays), began five out, and had uninterrupted play up until the first shot attempt. For each possession, footage was collected across the entirety of the possession, generating a curve of spacing vs. frame number. The trapezoidal rule was then used to approximate an area under this curve, generating a measure of ”total” spacing during the play. Finally, these ”total” spacing measures were averaged over all possessions with the given lineup, associating to each lineup a number representing how well they spread the floor.



Note: The area units are somewhat arbitrary. They have no correspondence with real-world distances.

The Mitchell, Allen, Strus, Garland and Mobley lineup yielded the best spacing out of the three tested. The fact that the second lineup - which only varies from the first lineup in the replacement of Caris LeVert by Darius Garland - has significantly better spacing demonstrates Darius Garland’s ability to draw defenders towards the perimeter.

Such an exercise with individual players can be reproduced to generate individual ratings of ”spacing produced”. The idea is to find two lineups, one with player X, the other with an average 3-pt-shooting replacement player. The change in spacing between those two lineups is then a measure of the impact of player X on spacing, roughly corresponding to ”how much player X expands/shrinks the size of the driving lane compared to an average 3-point-shooting player?”.

6 Further work

- Train the YOLOv8 model further on a basketball-unique dataset to allow better player identification on screens.
- Train the YOLOv8 model to identify the basketball
- Use homography to approximate flat planar coordinates for each player
- Instead of using the convex hull of the players, use the half circle bounded by their locations and the baseline.