Towards a Computer Vision Approach to Quantify Player Spacing in the NBA

Martin Bogaert mbogaert@mit.edu
Lucas Leforestier lucaslfr@mit.edu
MIT - 6.8300 Advances in Computer Vision

Abstract

While many NBA statistics aim to estimate player performance, none quantify the spacing provided by players on the court. Here we develop a computer vision approach to infer the position of players from broadcast video using object detection, player classification, and homography projection models. Subsequently, we introduce the adJusted Offensive Elongation metric, quantifying a player's ability to attract defenders. The developed pipeline shows promising results on five different sequences. It produces an average error of 6.4 feet on five known shot locations. Lastly, it finds that proficient shooter Stephen Curry is twice as tightly guarded than forward Draymond Green. This valuable information enables teams to refine their court spacing strategies and scout for valuable shooting specialists.

1. Introduction

Recently, novel statistics have been introduced to try to narrow down the skill-level of players in the National Basketball Association (NBA) to a single number. Furthermore, three-point shooting and spacing has become central to success in basketball. In the last two decades, the number of three-points shots attempted per season has tripled. However, while three-point percentage provides a good indication of a player's shooting prowess, it does not account for how tightly guarded a player is.

The project's aim is to develop the adJusted Offensive Elongation (**JOE**), a data-driven metric that quantifies how an offensive player attracts surrounding defenders. Moreover, most basketball organizations are much less technologically advanced than the NBA. This work thus provides a cheap alternative for leagues and teams to aggregate positional game data from broadcast videos.

This project leverages modern computer vision methods to quantify how much a player's shooting threat attracts defenders and thus opens up space in other areas of the court. It introduces a pipeline which takes a game broadcast video as an input and returns trajectory data for all players, utilizing detection, classification, and projection techniques.

2. Related work

The literature on player detection and tracking in sports introduces novel methods that aim to tackle challenges relating to video inputs and player identification. The most relevant research revolves around basketball and soccer games. This involves differentiating between multiple players and handling videos inputs as a sequence of frames [1] [2]. A share of the literature aims to use detection and tracking as complementary information to produce automated pipelines [3] [4].

Moreover, it proposes techniques to apply modern computer vision techniques to project positions and perform three-dimensional modelling of the ball and players from video input [5] [6]. It also aims to optimally estimate the camera's parameters and pose from the limited information provided by a one-camera system [7] [8].

Lastly, the literature proposes multiple valuable downstream analysis of the data, similar to the end-goal of this project. This includes analysis of patterns of play and tactics in basketball games [9] [10] [11], aiming to provide insights from the dynamics of player movements that are hard to discern with from a naked eye. Another avenue of analysis is to classify sequences of play in a game to segment a broadcast video [12].

3. Methods

3.1. Scope

To provide a proof of concept for the approach and limit the scope of the project, the latter focuses on the famous game 7 of the 2016 NBA Finals between the Cleveland Cavaliers and the Golden State Warriors which can be found in full length on YouTube [13]. From this game's video, a total of five sequences were selected to evaluate the pipeline results and compute the attraction metric. The sequences exclusively consider offensive plays by the Warriors and were chosen to limit the list players that the detection model must classify from. To build the training set used in the player classification section, the project uses a video of highlights of game 5 of the NBA Finals [14]. To evaluate the output trajectories, it uses a dataset of known shot locations [15],

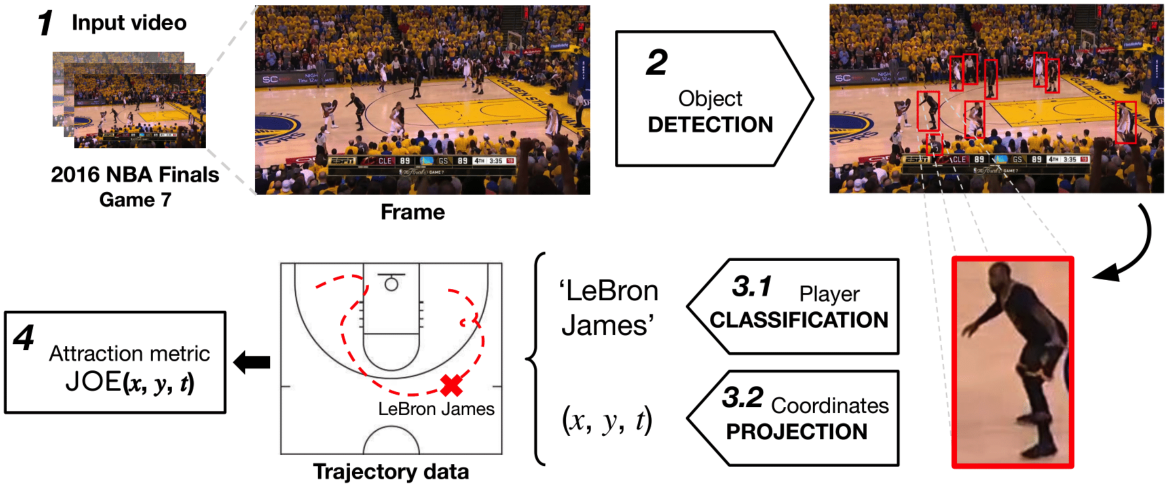

Figure 1. Graphical summary of the project's pipeline. It processes an input video sequence through detection, classification and projection models in order to produce a trajectory dataset that computes JOE for each offensive player.

allowing to cross-check with the produced trajectory data at the time of the shot.

Figure 1 presents the project's pipeline and main building blocks. From an input video, each frame is ran through an object detection model to identify players. Each player is then classified and its position is projected onto the (x, y) plane. This produces trajectory data for each player that is used as the input of the metric function.

3.2. Object detection

The key component of the project is the object detection model. In fact, the latter is needed to detect the players in the video, classify them, and project their position on the court. For this task, the project uses Hugging Face's pre-trained object detection transformer model [16]. The latter uses a DEtection TRansformer (DETR) model which utilizes an encoder-decoder architecture to perform detection tasks [17]. This detection model is used to build a training set of images and then within the pipeline to detect players in each frame of the sequences of interest. Since the model provides a label for each detected object, all objects which are not labelled as *person* are discarded.

3.3. Player classification

Following the detection of objects on the frames of the video, the central challenge is to classify them. Since there exist several techniques to do so, an ensemble approach was here utilized to perform the classification. The approach considers both image classification and number detection.

3.3.1 Convolutional neural networks

The first component of the approach is traditional image classification. Convolutional neural networks (CNN) were used to differentiate between the objects identified by the detection model. It was found that developing several coarser model was more efficient than using a global general model. The CNN pipeline thus uses a total of four models. First, in the broadcast video, there are three main types of detectable persons: players, referees, and fans. Therefore, the first CNN model does classification between these three classes. Detected referees and fans are then discarded. To constraint the final classification, the second CNN performs binary classification between Warriors or Cavaliers players. Lastly, two classification models were trained to classify among the five players of each team.

3.3.2 Number detection

The second building block of the classification approach involves scanning the input image to potentially read the number on the back of a player's jersey. The project uses EasyOCR [18], a pre-trained model that detects text on images. From an input image, it first detects if the player's number is visible. If it is, it returns the text on the image. Even though the player's jersey number is not always shown, this technique is very efficient and reliable to classify players whenever the number is visible. Whenever a number is detected, it maps this number to the corresponding list of players and the image may skip the CNN pipeline.

3.4. Coordinate projection

Once players are identified within the frames, the next step is to infer their position on the court. Since the video uses only one camera and there is no information available about its intrinsic and extrinsic parameters, triangularization cannot be used to compute the players' position. Therefore, the boundaries of the basketball court through the camera's point of view are used to compute an homography projection function that transforms the actual lines into lines seen from above. Once this function is defined, the same transformation can be applied to the position of the players.

The homography projection function relies on having perfect information of the court lines at every frame. However, in most frames, only two or three out of four lines are detectable (Figure 2). Therefore, since the sequences of interest are half-court plays, it is possible to hard-code the court delimitation. One possible enhancement is to infer the lines' positions according to where the basket and backboard are in the image by adding another object detection tailored for this task.

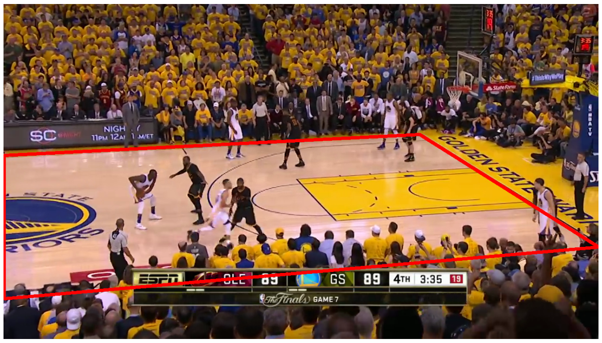

Figure 2. A frame with the court's delimitation shown

Finally, using the homography projection function specific to this problem, one can apply this transformation to get the modified frame. From these images, one can infer the position of the players' detection boxes in the plane of the court. Figure 3 shows an example of the projection of a frame to a bird's-eye view.

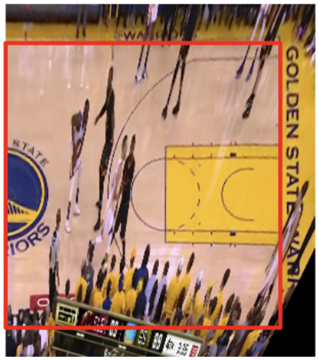

Figure 3. A frame projected to bird's-eye view plane

For some coordinates (x_{box}, y_{box}) corresponding to the bottom position of a detected player's box, the homography function is applied as followed:

$$\begin{bmatrix} X \\ Y \\ W \end{bmatrix} = \mathbf{H} \begin{bmatrix} x_{box} \\ y_{box} \\ 1 \end{bmatrix}$$

where $\mathbf{H}$ is the homography matrix. The corresponding bird's-eye view coordinates (x_{proj}, y_{proj}) are then inferred as:

$$\begin{bmatrix} x_{proj} \\ y_{proj} \end{bmatrix} = \begin{bmatrix} X/W \\ Y/W \end{bmatrix}$$

3.5. Trajectory interpolation

In the pipeline that flows from input video to trajectory data, there exist many potential sources of noise, ranging wrong player identification, erroneous coordinates projection, or missing object detection. The (x, y, t) data of a given player outputted by the model may therefore be noisy and not look like a smooth player movement trajectory. An algorithm that cleans the data for each player was thus devised. It first interpolates the missing coordinates from the surrounding time points. Then, points where the speed of a player must have exceeded a realistic threshold or that are outside the court's delimitation can be flagged as errors and discarded. A second interpolation can then be fitted to render a final trajectory. Overall, this smoothing procedure is efficient if most player's positions are accurately predicted.

3.6. Attraction function

Several approaches can be taken to compute an attraction function for the offensive players, i.e. **JOE**. Here a simple metric which computes the average distance of the closest defender at any time for a player i is used:

$$\texttt{JOE}_i = \frac{1}{T} \sum_{t \in [T]} \min_{j \in \mathcal{D}} ||\mathbf{x}_i(t) - \mathbf{x}_j(t)||$$

where T is the number of frames in the sequence, $\mathcal{D}$ is the set of defenders and $\mathbf{x}$ is a player's position vector.

4. Results

4.1. Player classification

From the hand-labelling of images from random frames of the game 5 video, the CNN models were trained and validated using around 1,000 samples. Table 1 summarizes the accuracies achieved by the four CNN models utilized in the player classification pipeline. The reported baseline systematically predicts the majority class. Moreover, Figure 4 present to confusion matrices for the classification models of players by team.

Model	Person	Team	Warriors	Cavaliers
No. classes	3	2	5	5
Baseline	0.79	0.54	0.39	0.36
Train	0.98	1.0	0.98	0.93
Test	0.97	1.0	0.93	0.89

Table 1. Model accuracies on the training and testing sets. **Person:** player, referee or fan classification; **Team:** binary player team classification; **Warriors:** Golden State Warrior player classification; **Cavaliers:** Cleveland Cavalier player classification

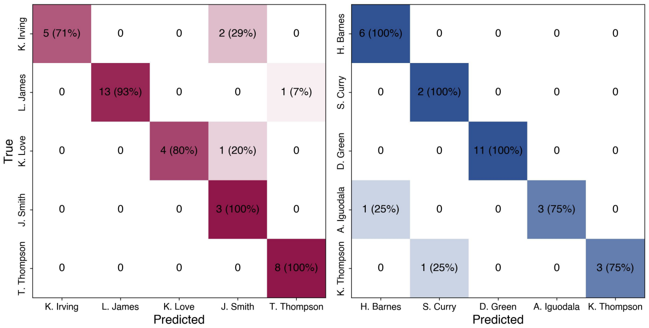

Figure 4. Confusion matrix for the Cavaliers and Warriors player classification neural networks

Following some trial-and-error with CNN architecture, the project converged on using three convolutional blocks for the person and team classifier networks, flattening into a 128 neurons-deep fully-connected layer. The Warriors and Cavaliers networks are slightly more complex, employing four convolutional blocks which flatten in a fully-connected layer of depth 256. All models were trained for 10 epochs using a batch size of 16 and do not seem to indicate signs of overfitting.

Regarding number detection, an image falls within one of three categories: a correct prediction, a false prediction or no prediction, as exemplified in Figure 5. From a set of about a hundred player images in which the number is potentially distinguishable, these categories occur at a rate of 25%, 10%, and 65%, respectively. However, most false predictions are easily discarded and changed to no prediction since there is only a small subset of possible numbers to classify from when the player's team is known. While the rate of accurate number readings is relatively low, it is still useful to make multiple high-confidence predictions in a sequence.

4.2. Trajectories projection

For any given frame, every point can then be projected using the homography projection function to match the court's delimitation from the bird's-eye point of view. Then, the player's raw trajectory is processed by the smoothing and errors discarding algorithms to produce a realistic player's trajectory over the sequence. Figure 6 shows the smoothed trajectory of a player during an example sequence.

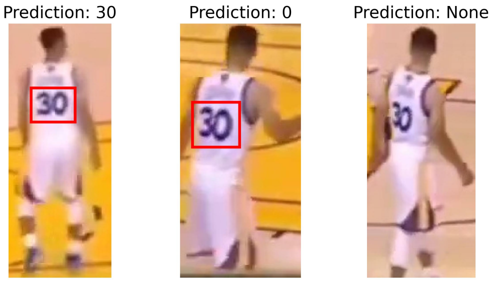

Figure 5. Example of EasyOCR predictions. From left to right: correct prediction, false prediction, and no prediction

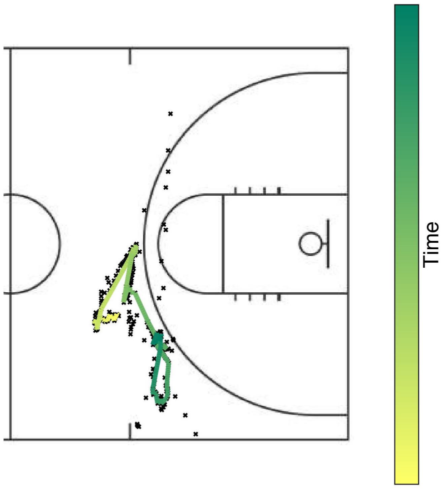

Figure 6. Example of the smoothed trajectory of LeBron James during a sequence of play. The black crosses correspond to the raw data points produced by the pipeline. It shows that points which entice excessive speed are discarded, yielding a realistic path.

For every selected sequence, the trajectories are evaluated by computing the difference between the projected and true shot locations. Due to time constraints, with the process being time-consuming and the fact that some elements (like court delimitation) of the pipeline needed to be hardcoded, only five shot locations could be tested for evaluation purposes. As a baseline, the center of the half-court is taken to predict the position of the shot.

As shown in Table 2 the results significantly outperform the baseline, both in terms of mean absolute error but also in terms of standard deviation, suggesting that the errors remain stable. Figure 7 presents an example of the error on one of the sequence.

	Baseline	Projected trajectory
MAE (ft)	12.1	6.4
Std. dev. (ft)	6.7	3.5

Table 2. Performance of the pipeline compared to a baseline

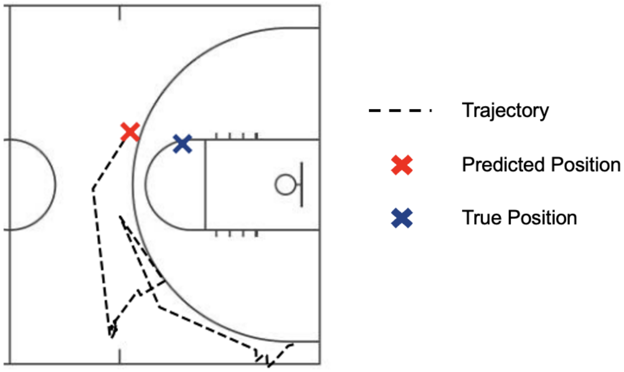

Figure 7. Predicted and true position of a shot by Klay Thompson over its trajectory during a sequence

4.3. Attraction metric

Using the attraction function defined above, the scores can be computed for any given sequence as their magnitude rely on what is happening on the court. Figure 8 shows the **JOE** scores for the five Warriors players on the court in one of the considered sequence. In this sequence, Draymond Green is left alone on the play without any defenders closing him out significantly. On the other hand, Andre Iguodala and Stephen Curry – who are know to be proficient shooters – are being defended more tightly, translating in lower **JOE** values.

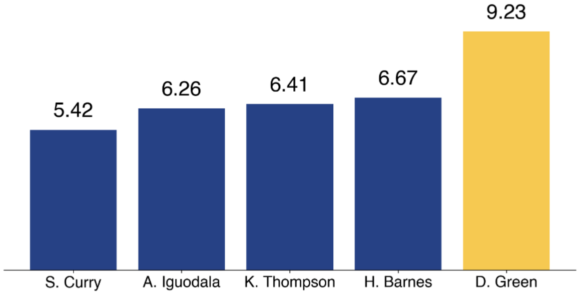

Figure 8. **JOE** metric in one of the sequence: mean distance to the closest defender (in feet) for every Warriors player

5. Conclusions and discussion

While the pipeline's limitations impacted the scalability of the model, the fact that it was only evaluated on a few sequences makes it hard to draw confident conclusions on accuracy. However, the pipeline still produced promising results as the trajectories and **JOE** metric is able to capture the relevant information of a given sequence.

The next steps of the project encompass three steps. First, to improve the whole pipeline for player classification. From trying more sophisticated models to increasing the number of the training samples, the ideal end result would to recognize any players from the pool of NBA athletes with a near perfect accuracy. However, labeled images taken from videos of better resolution would be required. Moreover, tracking and adding the ball trajectory could bring additional value to the pipeline. It would allow to create more complex attraction metric functions by differentiating between when a player is holding the ball (and attracting defenders more strongly) or not.

Second, for trajectory projection, it would require to evaluate the model on more sequences. This can be done with a refined pipeline that automates half-court sequence and court lines recognition. This approach can be first tested on easier examples, such as games in empty stadium in which all game lines are discernable at all times.

Lastly, an enhancement is to introduce and test different approaches for the attractiveness **JOE** metric, increasing in complexity. For example, a more sophisticated approach inspired by Coulomb's law can be defined. It would assign a charge to each player (attractive for offensive players; repulsive for defensive players) and thus define a force of attractiveness according to the distance of the players on the field from a player of interest. Subsequently, it is believed that statistically significant spacing patterns need to be observed from a high number of samples. It would thus need to average the score of players across a wider range of games to potentially observe any patterns in attractiveness from elite shooters in the NBA.

Work allocation

Martin: developed the player classification pipeline, from the use of the object detection model, to producing the training set, to adding up different layers such as team, number and player detection CNNs that were fine-tuned and enhanced throughout the project. He specifically ran all accuracies and metrics for all the neural network models.

Lucas: produced the coordinate and trajectory projection model, added smoothing and interpolation for better results, evaluated the positions and computed the attraction metric. He specifically ran all the trajectories for every player on the five selected sequences, and calculated the error on the shooter's projected position compared to the baseline.

References

[1] X. Fu, K. Zhang, C. Wang, and C. Fan, "Multiple player tracking in basketball court videos," *Journal of Real-Time Image Processing*, vol. 17, no. 6, pp. 1811–1828, 2020. 1

[2] C. Yang, M. Yang, H. Li, L. Jiang, X. Suo, L. Mao, W. Meng, and Z. Li, "A survey on soccer player detection and tracking with videos," *The Visual Computer*, 2024. 1

[3] Y. Pandya, K. Nandy, and S. Agarwal, "Homography based player identification in live sports," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 5209–5218, 2023. 1

[4] P. K. Santhosh and B. Kaarthick, "An automated player detection and tracking in basketball game," *Computers, Materials & Continua*, vol. 58, no. 3, pp. 625–639, 2019. 1

[5] Y. Ohno, J. Miura, and Y. Shirai, "Tracking players and estimation of the 3d position of a ball in soccer games," in *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, vol. 1, pp. 145–148 vol.1, 2000. 1

[6] A. Yamada, Y. Shirai, and J. Miura, "Tracking players and a ball in video image sequence and estimating camera parameters for 3d interpretation of soccer games," in *2002 International Conference on Pattern Recognition*, vol. 1, pp. 303–306 vol.1, 2002. 1

[7] L. Citraro, P. Márquez-Neila, S. Savarè, V. Jayaram, C. Dubout, F. Renaut, A. Hasfura, H. Ben Shitrit, and P. Fua, "Real-time camera pose estimation for sports fields," *Machine Vision and Applications*, vol. 31, no. 3, p. 16, 2020. 1

[8] M. C. Hu, M. H. Chang, J. L. Wu, and L. Chi, "Robust camera calibration and player tracking in broadcast basketball video," *IEEE Transactions on Multimedia*, vol. 13, no. 2, pp. 266–279, 2011. 1

[9] C. Tian, V. De Silva, M. Caine, and S. Swanson, "Use of machine learning to automate the identification of basketball strategies using whole team player tracking data," *Applied Sciences*, vol. 10, no. 1, 2020. 1

[10] J. Sampaio, T. McGarry, J. Calleja Gonzalez, S. Saiz, X. Alcázar, and M. Balciunas, "Exploring game performance in the national basketball association using player tracking data.," *PLoS ONE*, vol. 10, 01 2015. 1

[11] E. Santos-Fernandez, F. Denti, F. Mengersen, and A. Mira, "The role of intrinsic dimension in high-resolution player tracking data – insights in basketball," 2020. 1

[12] T. Facchinetti, R. Metulini, and P. Zuccolotto, "Filtering active moments in basketball games using data from players tracking systems," *Annals of Operations Research*, vol. 325, no. 1, pp. 521–538, 2023. 1

[13] ESPN, "[FULL GAME] Cleveland Cavaliers vs. Golden State Warriors — 2016 NBA Finals Game 7 — NBA on ESPN," 2020. 1

[14] NBA, "Warriors vs Cavaliers: Game 5 NBA Finals - 06.13.16 Full Highlights," 2016. 1

[15] Kaggle, "NBA Shots Dataset (2000 - Present)," 2022. 1

[16] H. Face, "Object detection," 2024. 2

[17] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," 2020. 2

[18] JaidedAI, "EasyOCR," 2023. 2