Single-Camera Basketball Tracker through Pose and Semantic Feature Fusion

Conference Paper · June 2019

DOI: 10.5281/zenodo.3346759

3 authors, including:

Adrià Arbués Sangüesa
University Pompeu Fabra
15 PUBLICATIONS **70** CITATIONS
SEE PROFILE

C. Ballester
University Pompeu Fabra
86 PUBLICATIONS **5,666** CITATIONS
SEE PROFILE

Single-Camera Basketball Tracker through Pose and Semantic Feature Fusion

Adrià Arbués-Sangüesa[1,2], Coloma Ballester[1], Gloria Haro[1]
[1] Universitat Pompeu Fabra (Barcelona, Spain) and [2] FC Barcelona

Abstract—Tracking sports players is a widely challenging scenario, specially in single-feed videos recorded in tight courts, where cluttering and occlusions cannot be avoided. This paper presents an analysis of several geometric and semantic visual features to detect and track basketball players. An ablation study is carried out and then used to remark that a robust tracker can be built with Deep Learning features, without the need of extracting contextual ones, such as proximity or color similarity, nor applying camera stabilization techniques. The presented tracker consists of: (1) a detection step, which uses a pretrained deep learning model to estimate the players pose, followed by (2) a tracking step, which leverages pose and semantic information from the output of a convolutional layer in a VGG network. Its performance is analyzed in terms of MOTA over a basketball dataset with more than 10k instances.

Keywords—Basketball, Deep Learning, Feature Extraction, Single-Camera, Tracking.

I. Introduction

Basketball professional European courts are 28 meters large and 14 meters wide, and 10 players (plus 3 referees) interact on it following complex patterns that help them accomplishing their goal, which can be either scoring or preventing the other team to score. These tactical plays include several kinds of movement, which might involve players moving together and close to each other, thus generating space and advantages. Given this scenario, and being aware that there is not an established multi-camera array set in these courts for tracking purposes (because of its cost and the low height of stadiums), broadcasting cameras are the main source of basketball video content. Usually, these cameras are set in the middle of the court in the horizontal axis, and camera operators just perform some panning or zooming during the game right at the same spot.

Given this video feed, a tracking-by-detection algorithm is adopted: first, potential players are detected, and then, features are extracted and compared, quantifying how much do players in different frames resemble. Then, players are tracked by obtaining a matrix with all similarities and minimizing the total cost of assignments. Several kind of features for establishing the similarities are evaluated:

- Geometrical features, which might involve relative distances (in screen-coordinates and expressed in pixels) between detected objects.
- Visual features, which may quantify how different boxes look alike by comparing RGB similarity metrics in different small neighborhood patches.
- Deep learning features, which can be obtained by post-processing the output of a convolutional layer in a Deep Neural Network.

Besides, we show that the combination with classical Computer Vision techniques helps improving the trackers' overall performance. In particular, camera stabilization based on homography estimation leads to camera motion compensated sequences where distances of corresponding players in consecutive frames are considerably reduced.

The aim of this paper is to prove that deep learning features can be extracted and compared with ease, obtaining better results than with classical features. For this reason, an ablation study for different tests in a given scenario is included. The remaining article is distributed as follows: in Section II related works are described. Later on, in Section III, the presented methods are detailed, involving the main modules of player and pose detection, feature extraction and matching; moreover, camera stabilization techniques and pose models are considered. Results are shown and discussed in Section IV, and conclusions are extracted in final Section V.

II. Related Work

Multi-object tracking in video has been and still is a very active research area in computer vision. One of the most used tracking strategies is the so-called tracking by detection, which involves a previous or simultaneous detection step [12], [15], [9], [6], [5]. Some of these works use a CNN-based detector with a tracking step [15], [6], while others are based on global optimization methods. Among them, a joint segmentation and tracking of multiple targets is proposed in [12], while in [9] a full-body detector and a head detector are combined to boost the performance. The authors in [5] combine Convolutional Neural Networks (CNNs) and a Temporal-Flow-Fields-based method. Another family of tracking methods which achieve a good compromise between accuracy and speed are based on Discriminant Correlation Filters (DCF). They are based on a first stage where features are extracted and then correlation filters are used. Initially, hand-crafted features like HoG were used and later on different proposals use deep learning features extracted with pretrained networks (e.g. [13]). The results are improved when learning the feature extraction network in an end-to-end fashion for tracking purposes [21]. The latest trend is to train deep learning based tracking methods in an unsupervised manner [22], [23].

On the other hand, pose tracking refers in the literature to the task of estimating anatomical human keypoints and assigning unique labels for each keypoint across the frames of a video [11], [10].

This paper addresses the problem of tracking basketball players in broadcast videos. This is a challenging scenario

where multiple occlusions are present, the resolution of the players is small and there is a high similarity between the different instances to track, specially within the same team members. For a deeper review of players detection and tracking in sports the interested reader is referred to the recent survey [20]. The authors of [18] also consider basketball scenarios seen from a broadcast camera and they deal with player identification. For that, they propose to use CNN features extracted at multiple scales and encoded in a Fisher vector.

III. Proposed Method and Assessment

In this Section, the implemented tracking-by-detection method is detailed. The associated generic pipeline can be seen in Figure 1 and it follows the subsequent stages:

A. For each frame, the basketball court is detected, with the purpose of not taking fans and bench players into account in the following steps. Also, a camera stabilization step may be included, and a discussion about its need in order to perform multi-tracking by reducing distances of objects within frames is provided.
B. Players are detected, together with their pose using a pretrained pose model, and bounding boxes are placed around all of them.
C. Features are extracted from these bounding boxes in combination with pose information. Several choices are analyzed in terms of features to be extracted.
D. By comparing features of all players in three consecutive frames (indicated by Frame N, N-1 and N-2, respectively, in Figure 1) and using a customized version of the Hungarian algorithm, tracking associations are performed.

A. Pre-Processing

1) Court Detection: Although court detection is not the main contribution of this research, the identification of visible court boundaries in the image is basic in order to filter out those candidates that are not taking part of the game actively (such as bench players or referees). It has to be mentioned that the basic filtering to be performed is thought for the vast majority of European courts, where court surroundings usually share the same color, and fans sit far from team benches. Knowing that in the broadcasting images the court results in a trapezoid with some visible boundaries, line segments are detected by using a fast parameter-less method based on the *a contrario* theory [7] (code available in [8]). Right after, segments with the same orientation and intersection at the boundaries of the image, as seen in Figure 2, are joint and considered as part of the same line; the dominant orientation will be considered as the one with the longest visible parts (proportional to the sum of individual segments' length). However, given that basketball courts have many parallel lines (such as sidelines, corner three line, paint sides,...), several line candidates have to be tested in order to find the real court surroundings. Moreover, two dominant orientations are taken into account: (1) the ones belonging to sidelines (intersections at both left-right image boundaries), and (2) the one belonging to the visible baseline (both baselines cannot be seen at the same time if the camera shot is an average one). Given the non-complex scenario of European courts, color filtering is used in the HSV colorspace by checking contributions all over the image; in the case of Figure 2, court surroundings are blue and the court itself is a bright brown tonality. For a given dominant orientation, the subsquent steps are followed:

1) First, a line candidate with the dominant orientation is set at the top (in the case of sideline candidates) / left side (baseline candidates) of the image.
2) Then, two parallel lines are set at a ± 25 pixel distance with respect to the line candidate.
3) Later on, and taking only the pixels comprised between the candidate line and the two parallel ones, the number of pixels that satisfy color conditions is computed for both sides independently. This is, if the candidate line is a potential sideline, the number of pixels is computed above and under it; instead, if the candidate line is a potential baseline, the number of pixels is computed at its left and its right. In the case of Figure 2, pixels with a Hue value between 120 and 150 degrees are the ones satisfying filter conditions.
4) The line candidate is moved 12 pixels towards the bottom (sidelines) / right side (baseline) of the image. The same procedure being followed in Steps 2 and 3 is applied again.
5) Once examined all possible cases, the line candidate with the maximum difference between above-below/left-right sides is set as the court limit. As it can be seen in the given example, the best court delimiters are the lines that stay right in the limit between brown-blue regions.

2) Camera Stabilization: In order to ease the tracking of the players, an additional camera stabilization step to remove the camera motion can be incorporated. Taking into account that its inclusion represents extra computations, in this paper, an ablation study is provided to discuss the extend of its advantages. When enclosed, the camera stabilization method and implementation in [17] is used. It estimates a set of homographies, each of which associated to a frame of the video and allowing to stabilize it. Table V in Section IV presents the quantitative results including it.

B. Player Detection

As mentioned in Section I, the presented tracker is based on multiple detections in each individual frame. More concretely, the implemented method relies on pose models techniques [14], [24], [2] stemming from an implementation of the latter [3]. Basically, this method is a bottom-up approach that uses a Convolutional Neural Network to: (1) detect anatomical keypoints, (2) build limbs by joining keypoints, and (3) merge limbs in the visible person skeleton. Given a basketball frame, the output of the main inference pose function is a 25×3 vector for each player, with the position (in screen coordinates) of 25 keypoints, which belong to the main biometric human-body parts, together with a confidence score. Note that there might be situations where specific parts might not be detected, resulting in unknown information in the corresponding entry

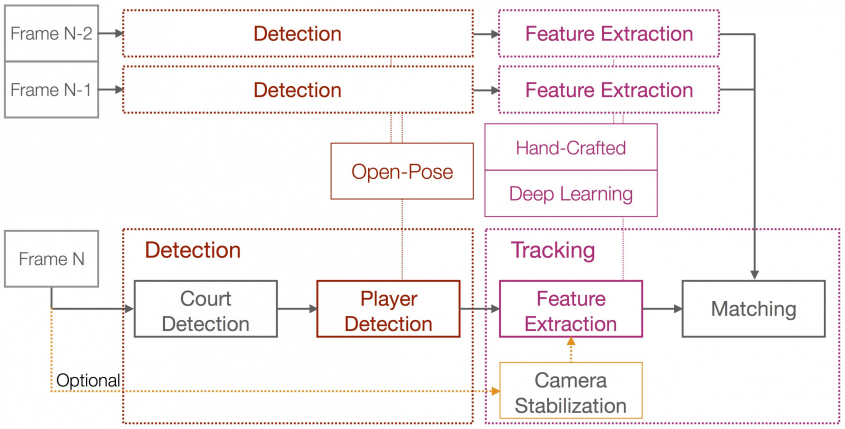
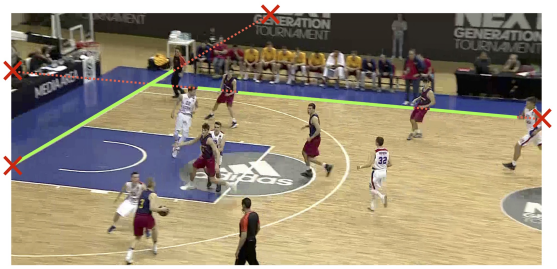


Fig. 1. Generic Pipeline: for each frame, players are detected (through pose models) and tracked (via feature extraction and matching).



(a)

(b)

Fig. 2. Court detection. (a) Different segments with the same orientation and intersections are joint; (b) Final segmentation result.

of the pose vector of the whole skeleton. In addition, 26 heatmaps are returned, indicating the confidence of each part being at each particular pixel. By checking all the parts' positions and taking the minima-maxima XY coordinates for each detected player, bounding boxes are placed around the respective players.

C. Feature Extraction

Once bounding boxes are obtained, their comparison must be performed in order to assign individual tracks for each box in time. With the purpose of quantifying this process, different approaches can be used whilst extracting features. In this subsection, all tested features used *a posteriori* are explained. For the remaining part of this subsection, B_{t_1} and B_{t_2} are considered as two different bounding boxes, detected at t_1 and t_2 respectively.

1) Geometrical Features: This classical approach can be used to measure distances or overlapping between bounding boxes of different frames. If the number of frames per second the video feed is not low, it can be assumed that player movements between adjacent frames will not be large; for this reason, players can be potentially found at a similar position in screen coordinates in short time intervals, so the distance between bounding boxes' centroids can be used as a metric. That is, given $\mathbf{x}_{B_{t_1}}$ and $\mathbf{x}_{B_{t_2}}$ as the centroids of two bounding boxes, the normalized distance between centroids can be expressed as

$$C_d(B_{t_1}, B_{t_2}) = \frac{1}{\sqrt{w^2+h^2}} \|\mathbf{x}_{B_{t_1}} - \mathbf{x}_{B_2}\|, \qquad (1)$$

where w and h are the width and the height of the frame domain. Another similar metric that could be used is the intersection over union between boxes, but due to the fact that basketball courts are usually cluttered and players move fast and randomly, it is not useful for this paper's purposes.

2) Visual Features: Distances might help distinguish basic correspondences, but this simple metric does not take into account key aspects, such as the jersey color (which team do players belong) or their skin tone. For this reason, a color similarity could be implemented in order to deal with these situations. Moreover, in this specific case, knowing that body positions are already obtained, fair comparisons can be performed, where the color surroundings of each part will be only compared to the neighborhood of the same part in another bounding box. Nevertheless, it has to be taken into account that only the pairs of anatomical keypoints present or detected in both B_{t_1} and B_{t_2} (denoted here as $\mathbf{p}_1^k$ and $\mathbf{p}_2^k$, respectively) will be used for the computation. The color and texture of a keypoint can be computed by centering a neighborhood around it. That is, let $\mathcal{E}$ be a squared neighborhood of 3×3 pixels centered at $\mathbf{0} \in \mathbf{R}^2$. Then,

$$C_c(B_{t_1}, B_{t_2}) = \frac{1}{255|S|\,|\mathcal{E}|} \sum_{k \in S} \sum_{\mathbf{y} \in \mathcal{E}} \|I_{t_1}(\mathbf{p}_1^k + \mathbf{y}) - I_{t_2}(\mathbf{p}_2^k + \mathbf{y})\| \qquad (2)$$

where S denotes the set of mentioned pairs of corresponding keypoints detected in both frames.

3) Deep Learning Features: Deep Learning (DL) is a broadly used machine learning technique with many possible applications, such as classification, segmentation or prediction. The basis of any DL model is a deep neural network formed by many layers. These networks serve to predict values from a given input. Convolutional Neural Networks (CNN) are special cases in which weights at every layer are shared spatially across an image. This has the effect of reducing the number of parameters needed for a layer and gaining a certain robustness to translation in the image. Then, a CNN architecture is composed by several kinds of layers, being convolutional layers the most important ones, but also including non-linear activation functions, biases, etc. This type of layers computes the response of several filters by convolving with different image patches. The associated weights to these filters, and also the ones associated to the non-linear activation functions, are learnt during the training process (in a supervised or unsupervised way) in order to achieve maximum accuracy for the concrete aimed task. It is well known that the first convolutional layers will produce higher responses to low-level features such as edges while posterior layers correlate with mid-, high- and global-level features associated to more semantic attributes. Bearing in mind that training a model from scratch is expensive, researchers use pretrained models and their corresponding weights for their purposes, such as by fine-tuning the model (for instance by feeding the model with new data and changing or adapting the previously obtained weights accordingly).

In the presented experiments, the popular VGG-19 network [19] is used for feature extraction, initialized with weights trained with ImageNet dataset [4]. The original model was trained for image classification, and its architecture consists of 5 blocks with at least 2 convolutional layers, and 2 fully-connected layers at the end that will output a class probability vector for each image. The network takes as input a $224 \times 224 \times 3$ image, and the output size of the second convolutional layer of each block is shown in Table I.

	Width	**Height**	**Nº Filters**
b2c2	112	112	128
b3c2	56	56	256
b4c2	28	28	512
b5c2	14	14	512

TABLE I
OUTPUT SIZE OF VGG-19 CONVOLUTIONAL LAYERS. IN THE FIRST COLUMN, B STANDS FOR BLOCK NUMBER AND C STANDS FOR THE CONVOLUTIONAL LAYER NUMBER INSIDE THAT BLOCK.

In order to feed the network with an appropriate sized image, a basic procedure is followed as seen in Figure 3: considering that player boxes are usually higher than wider and having the center of the bounding box, its height H_{B_t} is checked. Then, a squared image of $H_{B_t} \times H_{B_t} \times 3$ is generated around the center of the bounding box; finally, this image is resized to the desired width and height (224 and 224, respectively). In this way, the aspect ratio of the bounding box content does not change.

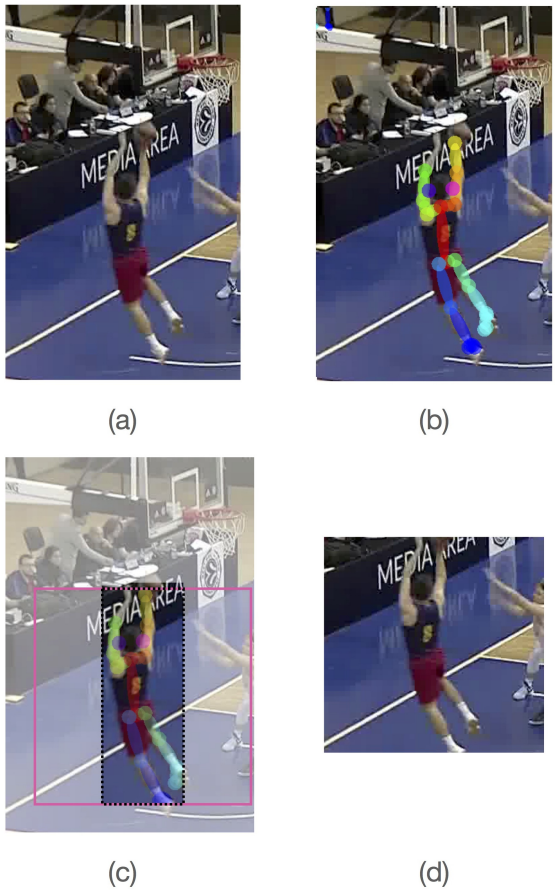


Fig. 3. Player and Pose Detection: (a) random patch of an image containing a player, (b) detected pose through pretrained models, (c) black: bounding box fitting in player boundaries, pink: bounding box with default 224×224 pixels resolution, (d) reshaped bounding box to be fed into VGG-19.

However, extracting deep learning features from the whole bounding box introduces noise to the feature vector, as part of it belongs to the background (e.g. court). Therefore, feature are only extracted in those pixels that belong to detected body parts, resulting in a quantized 1D vector with length equal to the number of filters. As detailed below, part positions have to be downscaled to the output size of the convolutional layer Moreover, all feature vectors must be normalized with L2 norm. An example using the 10th convolutional layer of VGG-19 is shown in Figure 4, where a $1 \times (25 \times 512)$ vector is obtained.

Once all boxes have their corresponding feature vectors, the metric defined in [22] is used to quantify differences; in particular, the similarity between two feature vectors $f_{t_1,k}^{y_{t_1}}$ and $f_{t_2,k}^{y_{t_2}}$, belonging to bounding boxes detected in t_1 and t_2 respectively, can be defined as:

$$Sim(f_{t_1,k}^{y_{t_1}}, f_{t_2,k}^{y_{t_2}}) = \frac{exp(f_{t_1,k}^{y_{t_1}} \dot{f}_{t_2,k}^{y_{t_2}})}{\sum exp(f_{t_1,k}^{y_{t_1}} \dot{f}_{t_2,k}^{y_{t_2}})} \quad (3)$$

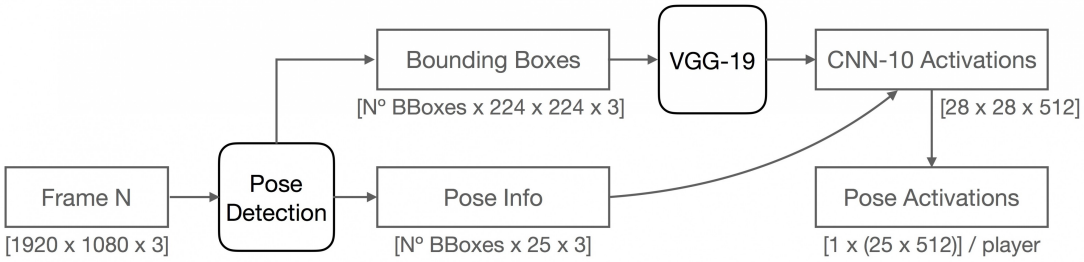


Fig. 4. Feature Extraction of all body parts using the 10th convolutional layer of a VGG-19 network.

where k corresponds to the particular body part and y_{t1} and y_{t2} to the pixel position inside the neighborhood being placed around the keypoint. Therefore, the total cost taking all parts into account is defined as:

$$C_{DL}(B_{t_1}, B_{t_2}) = \frac{1}{|S|} \sum_{k \in S} \max_{\substack{y_{t1} \in \mathcal{E} \\ y_{t2} \in \mathcal{E}'}} (Sim(f_{t_1,k}^{y_{t_1}}, f_{t_2,k}^{y_{t2}})) \quad (4)$$

where $\mathcal{S}$ corresponds, once again, to the set of detected parts in both frames, and $\mathcal{E}$ and $\mathcal{E}'$ correspond to the set of pixels in the neighborhood placed around each keypoint.

Nevertheless, two important remarks have to be pointed out:

1) Some of the Open Pose detected parts have low confidence. Given that, generally, there are more than 14 detected parts per player, all parts with lower confidence than 0.3 are discarded and not taken into account when extracting features. Hence, the subset $\mathcal{S}$ in Equations 2 and 4 considers all detected parts in both bounding boxes that satisfy the mentioned confidence threshold.
2) Convolutional layer outputs (as implemented in the VGG-19) decrease the spatial resolution of the input. Since non-integer positions are found when downscaling parts' locations (in the input image) to the corresponding resolution of the layer of interest, the features of the 2×2 closest pixels at that layer are contemplated. Then, the cost will be considered as the most similar feature vector to the 2×2 target one given. In Tables IV and V a discussion on the effect of the approximate correct location is included.

D. Matching

Having quantified all bounding boxes in terms of features, a cost matrix containing the similarity between pairs of bounding boxes is computed by combining the different extraction results. The suitability of the different types of features is evaluated by combining with appropriate weights them before building this matrix; in the presented experiments, the following weighted sum of different costs has been applied:

$$C(B_{t_1}, B_{t_2}) = \alpha C_{Feat1}(B_{t_1}, B_{t_2}) + (1-\alpha) C_{Feat2}(B_{t_1}, B_{t_2}) \quad (5)$$

where C_{Feat1} refers to C_d given by (1), C_{Feat2} refers either to C_{DL} in (4) or C_c in (2) and $\alpha \in [0, 1]$. From this matrix, unique matchings between boxes of adjacent frames are computed by minimizing the overall cost assignment:

1) For each bounding box in time t_N, the two minimum association costs (and labels) among all the boxes in t_{N-1} are stored in an $A_{t_N, t_{N-1}}$ matrix.
2) If there are repeated label associations, a decision has to be made:
   - If the cost of one of the repeated associations is considerably smaller than the others (by +10%), this same box is matched with the one in the previous frame.
   - If the cost of all the repeated associations is similar (less than 10%), the box with the largest difference between its first and second minimum costs is set as the match.
   - In both cases, for all boxes that have not been assigned, the label of their second minimum cost is checked too. If there is no existing association with that specific label, a new match is set.
3) In order to provide the algorithm with some more robustness, the same procedure described in steps 1 and 2 is repeated with boxes in t_N and t_{N-2}. This results in an $A_{t_N, t_{N-2}}$ matrix.
4) For each single box, the minimum cost assignment for each box is checked at both $A_{t_N, t_{N-1}}$ and $A_{t_N, t_{N-2}}$, keeping the minimum as the final match. In this way, a 2-frame memory tolerance is introduced into the algorithm, and players that might be lost for one frame can be recovered in the following one.
5) If there are still bounding boxes without assignations, new labels are generated, considering these as new players that appear on scene. Final labels are converted into unique identifiers, which will be later used in order to compute performance metrics.

IV. Results

In this section, a detailed ablation of quantitative results is provided and discussed, comparing all the above-mentioned techniques and combinations. Besides, the content of the gathered dataset is explained.

A. Dataset

A dataset of 22 European single-camera basketball sequences has been used. Original videos have full-HD resolution (1920×1080 pixels) and 25 frames per second, but in order to provide fair comparisons, only 4 frames are extracted

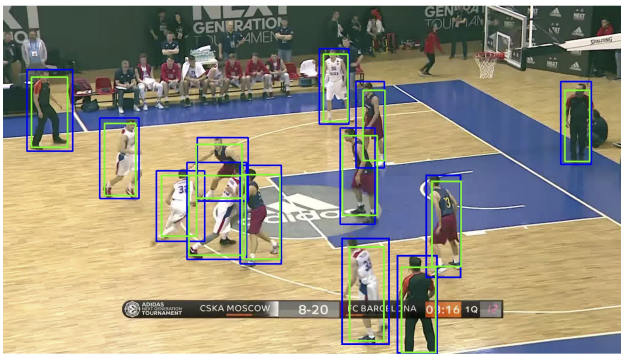


Fig. 5. Player Detections (green boxes) together with its ground truth (blue boxes).

per second. The included sequences involve static offensive basketball motion, with several sets of screens/isolations; moreover, different jersey colors and skin tonalities are included. However, the court is the same European one for all situations, and there are no fast break / transition plays, as in the case where all players run from one side to the other, due to the fact that camera stabilization techniques do not handle these situations. The average duration of these sequences is 11.07 seconds, resulting in a total of 1019 frames. Ground truth data is attached in the given dataset, containing bounding boxes over each player and all 3 referees (taking the minimum visible X and Y coordinates of each individual) in every single frame (when visible); this results results in a total of 11339 boxes.

B. Quantitative Results

Although it is not part of this article's contribution, quantitative assessment of the detection method is shown in Table IV-B and it is compared to the performance of the state-of-the-art YOLO network [16]; for a fair comparison, only the *person* detections within the court boundaries are kept in both cases. These detections can be seen in Figure 5 with their corresponding ground truth boxes.

	Precision	**Recall**	**F1-Score**
Open Pose	0.9718	0.9243	0.9470
YOLO	0.8401	0.9426	0.8876

TABLE II
DETECTION RESULTS

From now on, all quantitative tracking results will be expressed in the Multiple Object Tracking Accuracy (MOTA) metric, which is defined in [1] as:

$$MOTA = 1 - \frac{\sum_t fp_t + m_t + mm_t}{\sum_t g_t},$$

where fp_t, m_t, mm_t and g_t denote, respectively, to false positives, misses, missmatches and total number of ground truth boxes over all the sequence.

Another meaningful tracking metric that has been computed as well is the Multiple Object Tracking Precision (MOTP), which can be defined as:

$$MOTP = \frac{\sum_{i,t} IoU_{i,t}}{\sum_t c_t},$$

where $IoU_{i,t}$ and $\sum_t c_t$ correspond to the intersection over union between two boxes, and to the sum of correct assignations through the sequence, respectively. The detected bounding boxes for all the upcoming experiments are the same ones (thus the intersection with Ground Truth bounding boxes does not change neither), and knowing that the total number of instances is large, the MOTP results barely changes in all presented combinations of techniques: 0.6165 ± 0.0218.

Starting only with DL features (that is, $\alpha = 0$ in (5) and C_{Feat2} equal to C_{DL}), Table III shows the maximum MOTA metrics achieved after performing the extraction in the output of convolutional layers. As mentioned, a pretrained VGG-19 architecture is used, taking as an output the result of each second convolutional layer from the second to the fifth block. The best MOTA results are obtained with the output of the fourth block, corresponding to the 10th convolutional layer of the overall architecture. For the remaining tests, all DL features will be based on this layer, which has an output of size $28 \times 28 \times 512$.

Layer	b2c2	b3c2	b4c2	b5c2
MOTA	0.5396	0.5972	**0.6369**	0.6321

TABLE III
MOTA RESULTS OBTAINED WITH $\alpha = 0$ IN (5), C_{Feat2} EQUAL TO C_{DL} AND BY EXTRACTING FEATURES IN THE OUTPUT OF DIFFERENT CONVOLUTIONAL LAYERS.

Having tried all possible weights in 0.05 intervals, Table IV shows the most significant MOTA results for a non-stabilized video sequence. In this experiment, a comparison between Geometrical and DL features is shown, with the performance on their own as well as its best weighted combination. Besides, as explained in Subsection III-C3, when extracting DL features, three different tests have been performed regarding the neighborhood size. As it can be seen in Table IV, DL features outperform Geometrical ones, specially in the case of a 2x2 neighborhood. By combining them, and giving more weight to the DL side, results are improved in all cases, thus indicating that the two types of features complement each other. In Table V the same experiments are shown, but this time using a stabilizied video sequence. In this case, the Geometrical performance outperforms Deep Learning, but as it has been mentioned, these metrics will drastically drop if the included dataset sequences contain fast camera movements (or even large pannings).

From both Tables IV and V it can be deduced that the best filter size when extracting DL pose features is a 2x2 neighborhood. *A priori*, one might think that a 3x3 neighborhood should work better, as it is already including the 2x2 one, but a 3x3 spatial neighborhood in the output of the 10th convolutional layer is equivalent to a 24×24 real neighborhood around the specific part in the image domain. Accordingly, adding these extra positions will include court pixels in all

feature vectors, which might then produce a higher response in court-court comparisons, resulting in non-meaningful matches.

Neighborhood	α	1-α	MOTA
—	1	0	0.5689
1x1	0	1	0.5923
1x1	0.3	0.7	0.6289
2x2	0	1	0.6369
2x2	0.2	0.8	**0.6529**
3x3	0	1	0.6171
3x3	0.3	0.7	0.6444

TABLE IV
NON-STABILIZED RESULTS OBTAINED FROM ONLY 4 VIDEO FRAMES PER SECOND.

Neighborhood	α	1-α	MOTA
—	1	0	0.6506
2x2	0	1	0.6369
1x1	0.6	0.4	0.6752
2x2	0.55	0.45	**0.6825**
3x3	0.7	0.3	0.6781

TABLE V
STABILIZED RESULTS, WITH THE SAME 4 VIDEO FRAMES PER SECOND AND WEIGHTS AS IN TABLE IV.

Apart from comparing Geometrical and DL features through C_d and the mentioned different C_{DL}, the effect of Visual features (color similarity C_c, explained in Subsection III-C2) is checked too. In Table VI, the best weighted combinations in terms of MOTA are shown for a non-stabilized and a stabilized video sequence. In both cases, DL features outperform color ones by a 3% margin. The combination of all Geometrical, Visual, and DL features outperforms the rest of techniques but just by a 0.2%, which comes at a cost of computation expenses, so it is worth using only DL features.

Combination of Features	MOTA
Geometrical + Visual	0.6233
Geometrical + VGG	0.6529
Geometrical + Visual [Stab]	0.6583
Geometrical + VGG [Stab]	0.6825
Geometrical + VGG + Visual [Stab]	**0.6843**

TABLE VI
EFFECT OF VISUAL AND DEEP LEARNING FEATURES IN COMBINATION WITH GEOMETRICAL ONES.

In order to break down and evaluate the contribution in MOTA of every single pose part, Table VII is displayed; these results have been obtained with a 2x2 neighborhood around parts, and without combining with Geometrical features. As it can be seen, there are basically three clusters:

1) Discriminative features, above a 0.35 MOTA, that manage to track at a decent performance only with a 1×512 feature vector/player. These parts (shoulders, chest and hip) belong to the main shape of human *torso*, and it coincides with the jersey-skin boundary in the case of players.
2) Features that stay within a MOTA of 0.20 and 0.35, which are not tracking players properly but their contribution might help the discriminative ones to get higher performance metrics. These parts include skinned pixels of basic articulations such as elbows, knees, and ankles.
3) Concrete parts that have almost no details at a coarse resolution, thus resulting in low MOTA performance. Eyes could be an example: although people's eyes have many features that made them discriminative (such as shape, color, pupil size, eyebrow's length), players' eyes in the dataset images do not embrace more than a 5x5 pixel region, and all of them look the same shape and brown or darkish. This results in poor tracking results when checking only for these parts.

Part	MOTA
Chest	0.5349
L-Shoulder	0.4726
R-Shoulder	0.4707
R-Hip	0.3961
Mid-Hip	0.3956
L-Hip	0.3867
L-Knee	0.3156
R-Knee	0.3062
L-Elbow	0.2862
R-Elbow	0.2545
R-Ankle	0.2418
L-Ankle	0.2407
L-Toes	0.1935
R-Toes	0.1920
L-Ear	0.1348
L-Heel	0.1259
L-Wrist	0.1235
R-Heel	0.1126
L-Mid-Foot	0.1116
R-Wrist	0.1111
R-Mid-Foot	0.0964
L-Eye	0.0916
Nose	0.0771
R-Eye	0.0655
R-Ear	0.0677

TABLE VII
INDIVIDUAL PART TRACKING PERFORMANCE, OBTAINED WITH $\alpha = 0$ IN (5) AND C_{Feat2} EQUAL TO C_{DL}

Given the mentioned clusters, 3 different tracking tests have been performed taking only some parts into account, in particular, taking all the body parts that had a MOTA performance by itself higher than: (1) 0.35, (2) 0.20, (3) 0.10, belonging to (1) 6, (2) 12 and (3) 20 parts, respectively. Results are shown in Table VIII, where it can be seen that the second and third cluster complement the top ones, while the bottom-5 parts actually contribute to a drop in MOTA. The drawback of this clustering is that it requires some analysis that cannot be performed in test time, and different video sequences (*i.e* different sports) might lead to different part results.

A qualitative visual detection and tracking result (obtained with the best combination of Geometrical + Deep Learning features without camera stabilization) is displayed in Figure 6, where players are detected inside a bounding box, and its color indicates their ID; as it can be seen, all 33 associations

Part MOTA	Nº of Parts	Total MOTA
>0.35	6	0.6105
>0.20	12	0.6412
>0.10	20	**0.6423**
all	25	0.6369

TABLE VIII
CLUSTERING PART RESULTS ($\alpha = 0$ AND $C_{Feat2} = C_{DL}$)

are properly done except a missed player in the first frame and a missmatch between frames 2 and 3 (orange-green boxes)

V. CONCLUSIONS

In this article, a single-camera multi-tracker for basketball video sequences has been presented. Using a pretrained model to detect pose and humans, an ablation study has been detailed in order to address the feature extraction process, considering three types of features: Geometrical, Visual and Deep Learning based. In particular, Deep Learning features have been extracted by combining pose information with the output of convolutional layers of a VGG-19 network, reaching a maximum of 0.6843 MOTA performance. Several conclusions can be extracted from the presented experiments:

- In the case of VGG-19, DL features extracted from the 10th convolutional layer present the best accuracy; moreover, placing a 2x2 neighborhood around downscaled body parts improves the tracking performance.
- Classical Computer Vision techniques such as camera stabilization can help improving the Geometrical features performance, but it might have related drawbacks, such as the incapability of generalization to all kinds of camera movements.
- DL features outperfom Visual ones when combining with Geometrical information. The combination of all of them does not imply a performance boost.
- When extracting pose features from convolutional layers, those body parts that cannot be distinguishable at a coarse resolution have a negative effect in the overall performance.

Future work could involve the fine-tuning of a given network in order to get specific weights for tracking purposes. This training should be done in an unsupervised/self-supervised way, and a bigger dataset will be used, including different type of basketball courts and all kind of plays. Moreover, if there is no need to label ground truth data, this new model could be also trained with other sports' data, thus potentially creating a robust multi-sport tracker.

ACKNOWLEDGMENTS

The authors acknowledge partial support by MICINN/FEDER UE project, reference PGC2018-098625-B-I00, H2020-MSCA-RISE-2017 project, reference 777826 NoMADS and F.C. Barcelona's data support.

REFERENCES

[1] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: the clear mot metrics," *Journal on Image and Video Processing*, vol. 2008, pp. 1, 2008.
[2] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2017, pp. 1302–1310.
[3] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "OpenPose GitHub Repository" in https://github.com/CMU-Perceptual-Computing-Lab/openpose, last accessed May 18th 2019.
[4] J. Deng,and W. Dong, and R. Socher, and L. Li, and K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," CVPR09, 2009
[5] A. Doering, U. Iqbal, and J. Gall, "Joint flow: Temporal flow fields for multi person tracking," *arXiv preprint arXiv:1805.04596*, 2018.
[6] R. Girdhar, G. Gkioxari, L. Torresani, M. Paluri, and D. Tran, "Detect-and-track: Efficient pose estimation in videos," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2018, pp. 350–359.
[7] R. Grompone Von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "Lsd: A fast line segment detector with a false detection control," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 4, pp. 722–732, 2010.
[8] R. Grompone Von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "Lsd: a line segment detector," *Image Processing On Line*, vol. 2, pp. 35–55, 2012.
[9] R. Henschel, L. Leal-Taixé, D. Cremers, and B. Rosenhahn, "Fusion of head and full-body detectors for multi-object tracking," in *IEEE Conf. on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 1509–150909.
[10] E. Insafutdinov, M. Andriluka, L. Pishchulin, S. Tang, E. Levinkov, B. Andres, and B. Schiele, "Arttrack: Articulated multi-person tracking in the wild," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2017, vol. 4327.
[11] U. Iqbal, A. Milan, and J. Gall, "Posetrack: Joint multi-person pose estimation and tracking," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2017, pp. 2011–2020.
[12] A. Milan, L. Leal-Taixé, K. Schindler, and I. Reid, "Joint tracking and segmentation of multiple targets," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2015, pp. 5397–5406.
[13] Y. Qi, and S. Zhang, and L. Qin, and H. Yao, and Q. Huang, and J. Lim, and M.-H. Yang, "Hedged deep tracking," Proceedings of the IEEE conference on computer vision and pattern recognition, 2016
[14] V. Ramakrishna, D. Munoz, M. Hebert, J. Andrew Bagnell, and Y. Sheikh, "Pose Machines: Articulated Pose Estimation via Inference Machines," in *IEEE European Conf. Computer Vision*, 2014, pp. 33–47.
[15] V. Ramanathan, J. Huang, S. Abu-El-Haija, A. Gorban, K. Murphy, and L. Fei-Fei, "Detecting events and key actors in multi-person videos," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2016, pp. 3043–3053.
[16] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.
[17] J. Sánchez, "Comparison of motion smoothing strategies for video stabilization using parametric models," *Image Processing On Line*, vol. 7, pp. 309–346, 2017.
[18] A. Senocak, T.-H. Oh, J. Kim, and I. S. Kweon, "Part-based player identification using deep convolutional representation and multi-scale pooling," in *IEEE Conf. on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 1732–1739.
[19] K. Simonyan, and A Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014
[20] G. Thomas, R. Gade, T. B. Moeslund, P. Carr, and A. Hilton, "Computer vision for sports: current applications and research topics," *Computer Vision and Image Understanding*, vol. 159, pp. 3–18, 2017.
[21] Q. Wang, and J. Gao, and J. Xing, and M. Zhang, and W. Hu, "Dcfnet: Discriminant correlation filters network for visual tracking," arXiv preprint arXiv:1704.04057, 2017
[22] X. Wang, and A. Jabri, and A. Efros, "Learning Correspondence from the Cycle-Consistency of Time," arXiv preprint arXiv:1903.07593, 2019
[23] N. Wang, and Y. Song, and C. Ma, and W. Zhou, and W. Liu, and H. Li, "Unsupervised Deep Tracking," arXiv preprint arXiv:1904.01828, 2019
[24] S. E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional Pose Machines," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2016, pp. 4724–4732.

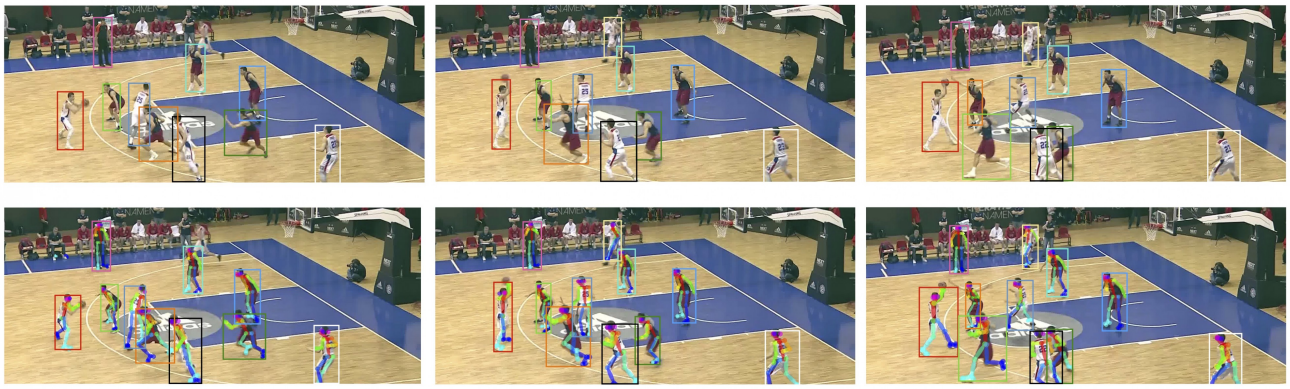


Fig. 6. Obtained tracking and pose results in three consecutive frames, where each bounding box color represents a unique ID.