Tactical Camera on Basketball Footages

Andrea Montibeller
University of Trento
Department of Information Engineering and Computer Science (DISI)
Trento, Italy
andrea.montibeller@studenti.unitn.it

I. Introduction

The studyies of the athletes statistical footage has become, day after day more important, and occupy nowadays a relevant position during the training phase. If since some years ago the teams were providing to the single players just the videos of their game so as they can analyse their errors, what the teams are looking for today is not just a video but also a tool so as the players can have a numerical and graphical reference to better understand where they have to improve.
Most of the biggest club have already implemented these algorithms and the purpose of this report is to explain how to, dealing with all the limitation of the case, implement something that goes as close as possible to the state of the art.
For this purpuse we dealt with problem related to the position of the camera, the distortion introduced by this one and the necessary calibration, the homography's problem in order to represent on a reference figure the trajectory of the athletes and the computation of the game parameters. Apart from this other problems concerned the research of a robust tracking algorithm and the implementation of a method able to solve the problem of the occlusions, that during a basketball game occur several times.
The report is organized as follow: **Section 2.** *Problem Formulation and Mathematical Modeling*, **Section 3.** *Precedent Work*, **Section 4.** *Improvement and Actual state of the Algorithm*, **Section 5.** *Experimental Results* and **Section 6.** *Conclusions*.

II. Problem Formulation and mathematical modeling

As already said in the introduction the research of algorithm that permits a better analysis and understanding of the movements, game's decisions and gap with respect to the other athletes has become day by day more important. Because the state of the art doens't present a general method that works perfectly for every scenario, the reaserch related to the best tracker and algorithm to use become really important.
Starting from the work done by another group of students, where their focus was in particular to develop an algorithm able to deal with the problem related to the camera calibration, the homography and the tracking of single players, our goal was generalize the method in order to being able to track more than one single player, dealing with the problem of the occlusions and collect their game parameters at the same time. To accomplish this goal, the precedent group already did an excellent job for the calibration and homography part.
Thus, first of all we tried to understand how to imporve the robustness of the tracker in case of occlusions, that occurs if an object that you are tracking is hidden (occluded) by another object (in our case, players occluded by other players). Another problem was given by the fact that, color based trackers suffer even more if they play in an enviroment where the targets are really big and the colors of their features are really similar not only to the features of the other objects but also with the ones of the background (i.e. piece of the basketball court really similar to the basketball's players uniforms). To solve this problem we opt for the use of a CSRT tracker [1] to whom we integrate a bayesian filter, in particular the Kalman one [2] and a re-initialization algorithm based on the analysis of the histogram's distribution of the first selected bounding box with respect to the predicted one, in order to increase the robustness of the multi-tracker.

A. Notation and Preliminaries

In this paper we will denote with the boldface font the vectors (or matrix) of length specified in the text. If $\mathbf{X}$ is an $M \times N$ matrix, with $\mathbf{X}_{(i,j)}$ we denote the *i,j*th element, while with $\mathbf{X}_{(i)}$ we denote the *i*th component of $\mathbf{X}$ when it is converted into a vector.
All the operations among vectors and matrices (i.e. product, ratio, raising to power, etc.)are *element wise*. The Euclidean dot product of vectors is denoted as $X \cdot Y$ and the Euclidean norm as $||X|| = \sqrt{X \cdot X}$.

III. Precedent Work

As explain in the precedent section the developed work is based on a precedent project.
The precedent group focused on the camera calibration and homography.
As explained in their presentation, the recording camera has been placed on a higher position on the stands in order to being able to record all the basketball court. Of course, for being able to do this the camera has been equiped with a fish-eye lens, that as consequence introduce a barrell distortion [3].
In order to avoid errors in phase of analysis of the players parameters due by the over mentioned distortion, what they did was, first of all, calibrate the camera.

A. Camera Calibration

During the camera calibration phase, the OpenCV calibration method has been used [4].
This calibration method, provided by OpenCV, only requires the shootings of the same calibration template for more than two images from different angles. After this part, the algorithm is able to obtain the internal and external camera parameters. The internal parameters are the camera ones such as the image center, focal length, lens distortion, etc. On the countrary, the external parameters returns information about the three-dimensional position and orientation of the camera coordinates system respect the world coordinates system.
Once that the dataset of images of the calibration template taken from different positions has been processed and the parameters extracted, the calibration is completed exploiting them and inverting the tangential and radial distortion using the following formulas.

$$\begin{aligned} \mathbf{x_{corrected}} &= x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6); \\ \mathbf{y_{corrected}} &= y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6); \end{aligned} \tag{1}$$

1: Formula for the radial correction.

$$\begin{aligned} \mathbf{x_{corrected}} &= x + [2p_1 xy + p_2(r^2 + 2x^2)]; \\ \mathbf{y_{corrected}} &= y + [2p_2 xy + p_1(r^2 + 2y^2)]; \end{aligned} \tag{2}$$

2: Formula for the tangential correction.

B. Homography

After the calibration part, they took care of finding a match between a point inside the video and the same one in the diagram of the basketball court.
To accomplish this goal they made use of the homography [5]. Because the homography is an operation that relates the transformation between two planes, it expressed as reported in **Formula 3**:

$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} \tag{3}$$

3: Homography Equation.

Where x_1 *and* y_1 are the coordinates of the first plane, x_2 *and* y_2 are the coordinates of the second one and the elementes of the matrix h_{ij}, with $i = 1, ..., 3$ and $j = 1, ..., 3$ are the components of the affine transformation to get the new points.

IV. Improvement and Actual state of the Algorithm

Altought the homography was well defined the tracking algorithm in case of occlusions had some problems following the target. One of the first thing that we noticed was that the trackers used in the precedent implementation were too weak and not robust enough for the application. As consequence, the first part of the research focused on the selection of the new tracker [6]. After had discard trackers as *Boosting*, *MIL* and *KCF* because of their time consuming or their low perfomance in case of occlusion or the *TLD*, *MOSSE* and *GOTURN* for their inaccuracy, we decided to opt for the *CSRT tracker*. Even if this tracker is slightly slower than the *KCF*, the *CSRT* presents better permance respect to the other trackers dealing particulary well also with the occlusions.
As written in [1], it exploits two simple standard features sets: HoGs and Color name, making use of discriminative correllation filters (DCF) with channel and spatial reliability.
The spatial reliability map is used to adapt the filter support to the part of the object suitable for tracking. This is quite important since overcomes all the problems related to the rectangular shape assumption of the target. It is estimated using the output of a graph labelling problem, solved for each frame.
The channel reliabity and its scores instead, are used for weighting the per-channel filter response in localization. Finally its estimation is done from the properties of the constrained least-square solution of filter design.
Once initialize this tracker with the first bounding box, the *update step* consist in:

- The training region is centered at the target location;
- The foreground and the background histogram are extracted and updated by an exponential moving average with a given learning rate. The foreground histogram is extracted by an Epanechnikov kernel within the bounding box and the background's one neighborhood that are twice the object size.
- The spatial reliability map is constructed and the optimal filters computed by optimizing.
- The per-channel learning reliability weights are estimated from the correlation responses and the current frame reliability weights are computed from detection and learning reliability.
- The filter and channel reliability weights are updated by the exponential moving average with given learning rate.

After this part, because of the problem due to the occlusions was not solved yet, we moved to the implementation of a *Kalman Filter* [2] [7] so as to avoid, as much as possible, the over mentioned problem. Kalman filters are nowadays, the most widely used variant of the *Bayesian Filter*.

They approximate beliefs by their first and second moment, which is virtually identical to unimodal Gaussian representation:

$$Bel(x_t) = \mathcal{N}(x_t;\mu,\Sigma)$$
$$= \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} exp\left[-\frac{1}{2}(x_t-\mu_t)^T\Sigma_t^{-1}(x_t-\mu_t)\right]; \quad (4)$$

4: Approximate beliefs.

Where $Bel(x_t)$ is the beliefs, μ_t the distribution's mean and Σ_t the covariance matrix.
The Kalman Filters are optimal estimators in case of gaussian's initial uncertainty and if the observation models and the system dynamics are linear functions. Because we were not always working under these hypothesis, playing with the transition, control and measurement matrix (being something OpenCV let us did) we extended the Kalman filter for systems that are not strictly linear [7]. Generally, as explained in [7] a system is linearized using the first-order Taylor series expansions.
The decision to implement a Kalman filter was given by their high efficiency even if this one comes at the cost of restricted computational power given to the necessity of low uncertainty in the state and by the fact that it can represent only unimodal distributions. Anyway, in our case it works pretty well since we know the target's initial location and so we can limit this uncertainty.
In the end, as written in the introduction we take care to re-initialize the tracker if the similarity between the histogram of the initial bounding box with respect to the predicted one is lower than 0.3 (30%). We do this in order to try improve the robustness of the tracker since the players move in different areas of the court having different colors and textures. The choice to set the threshold $\tau = 0.3$ has been computed empirically. We tested different values of the threshold on different target trying to find out the best trade off in order to re-initialize the system (if necessary) without that this operation was a cause of the loss of the target. During the tests we notice that the contribute of the *Kalman Filter* was way higher than the one provided by this re-initialization step, thus we choose to use it just as a "last chance" feature of our algorithm. As we show in **Table 1** and on our results, with $\tau = 0.3$ we re-initialize the tracker only in case of emergency avoiding re-initialization that would have lead us to errors. The table below shows how many times, with consequent loss of the targets showed after the table, the trackers have been re-initialize for a given threshold.

Thresholds	Updates Target1 Test1	Updates Target2 Test1	Updates Target1 Test2	Updates Target2 Test2
0.8	17	15	10	12
0.75	7	12	16	9
0.7	6	7	6	8
0.65	4	5	8	6
0.6	4	2	3	1
0.55	1	1	1	2
0.5	1	1	0	6
0.45	0	0	1	2
0.4	0	0	1	1
0.35	0	0	0	1
0.3	0	0	0	0

Table I: Times of unsuccesfull update for a given threshold.

Figure 1: Target1 (first on the left) and Target2 of Test1.

Figure 2: Target1 (first on the left) and Target2 of Test2.

In the end, after we solved the problem of the occlusions, because the precedent algorithm was not able to track more than one player per time we expanded the single tracker to multi-tracker and we add, to the already computable parameters (i.e. velocity, trajectory and length of the trajectory) the accelleration, being a very explosive sport.
Because the precedent algorithm was able to work only with video of 9 seconds and 25 fps we take care of the generalization of this part too. All the selected target with their own bounding box are showed in the video with different colors. The name of these colors are used in the text file, given as output at the end of the processing, to recognize the players and their parameters.

V. EXPERIMENTAL RESULTS

We tested our algorithm with 4 clip showing 4 different actions and many different game movements. For each of these clips we run more than one test, checking the performance in case of one, two ,three, four and five tracked players also to understand how big was the computational's cost. The results are presented in the **Table 2**.

N^o of Players	Mean Time to process 1 second of Video
1	2.6
2	4 sec.
3	5 sec.
4	6 sec.
5	7.5 sec

Table II: Times required to process one second of Video in relation with the selected number of players.

Even if the clips showed very different scenarios with different types of partial and total occlusion in the most of the case the algorithm answers good to these problems. In case of partial occlusion it was able to don't lose the tracked player and in case of total occlusion all depends by the dimension of the player and by the color of his uniform that, especially from this point of view of the camera, can create some problems. Below we show two particulary successfull results with their corresponding trajectory.

Figure 3: Multitracking of four players.

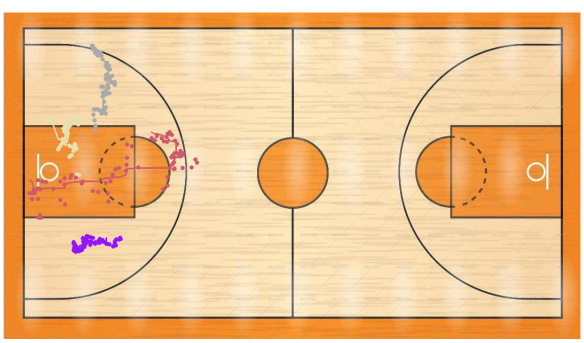

Figure 4: Trajectory of four players on the reference board.

Figure 5: Multitracking of five players.

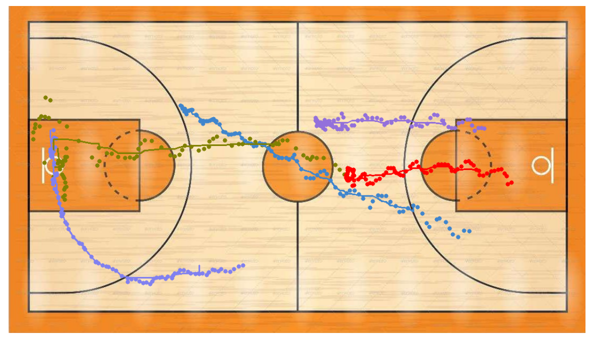

Figure 6: Trajectory of five players on the reference board.

Of course is not always easy obtain good results. A problem that we notice many time when we tried to track the players of the Zenit (i.e. white uniform) was that because the similarity between their uniform and the texture of some areas of the court, the tracker fails even in case of easy trajectory. A classical example is the one showed in **Figure 7** where, if the selection of the bounding box was not really precise happened that the tracker get stucked finishing with tracking some area representing the floor of the basketball court.

Figure 7: Multitracking of four players.

Thus the dependencies of this algorithm and this tracker to the selected bounding box (a bounding box respect to another one, even if similar can guarantee in some situation different results) is something that, in a future should be improved in order to return to the users a more robust and easier tool to use.

VI. CONCLUSION

After had run the algorithm on many video and different game's scenario we can be satisfy of the most of the results. The *Kalman Filter* solved the most of the problem related to the occlusions and the algorithm seems to be stable at the re-initialization process (i.e. it does not get stucked or lost in some part of the video).

About the time consuming, even in the case of tracking of five players, it remains in an acceptable range of time. What is also helping in this application is that the different actions are played in short time intervals that are easy to select and analyse one by one.

As mentioned before, a problem that was not solved is the one related to the confusion that may fools the tracker in case of players with uniforms too similar to the background. Trying to solve this problem could be an hot topic for future works because to this problem is also related the one concerning the bounding box selection that, in some case has to be really precise in order to help the tracker to regularly follow the target.

REFERENCES

[1] A. Lukezic, T. Vojit, L. C. Zajc, M. Kristan, "Discriminative Correlation Filter Tracker with Channel and Spatial Reliability, Springer Science+Business Media, LLC, part of Springer Nature 2018" 8 January 2018.

[2] G. Welch and Gary Bishop "An Introduction to the Kalman Filter," Department of Computer Science, University of North Carolina, June 2006.

[3] Wikipedia, "Distortion (optics)" https://en.wikipedia.org/w/index.php?title=Distortion_(optics)&oldid=870743290, 2018.

[4] Y. M. Wang, Y. Li, and J. B. Zheng, "A Camera Calibration Technique Based on OpenCV, College of Infonnatics and Electronics Zhejiang Sci-Tech University Hangzhou, China".

[5] OpenCv,"Basic concepts of the homography explained with code," https://docs.opencv.org/3.4.1/d9/dab/tutorial_homography.html.

[6] PyImageSearch,"https://www.pyimagesearch.com/2018/07/30/opencv-object-tracking/," https://www.pyimagesearch.com/2018/07/30/opencv-object-tracking/.

[7] D. Fox, J. Hightower, L. Liao,D. Schulz and G. Borriello, "Bayesian Filtering for Location Estimation,IEEE CS and IEEE ComSoc" 2003 .