As Seen on TV: Automatic Basketball Video Production using Gaussian-based Actionness and Game States Recognition

Julian Quiroga Henry Carrillo Edisson Maldonado John Ruiz Luis M. Zapata*
Computer Vision, Genius Sports
Medellín, Colombia
{julian.quiroga,henry.carrillo,edisson.maldonado,john.ruiz,luis.zapata}@geniussports.com

Abstract

Automatic video production of sports aims at producing an aesthetic broadcast of sporting events. We present a new video system able to automatically produce a smooth and pleasant broadcast of Basketball games using a single fixed 4K camera. The system automatically detects and localizes players, ball and referees, to recognize main action coordinates and game states yielding to a professional cameraman-like production of the basketball event. We also release a fully annotated dataset consisting of single 4K camera and twelve-camera videos of basketball games.

1. Introduction

Automatic video production of sports events is an interesting problem both from the commercial perspective [1] and from the computer vision side [34]. For the former, low latency streaming of auto-produced sports events is of interest for sports federations to growth their fan base, for team sponsors to get brand exposure and for fans to keep up to date with their team performance. Automatic video production is particularly relevant for niche sports events (e.g., Kenyan Basketball Premier League) where often there is not TV broadcast available. Regarding quality, automatic video production of sports events should produce an aesthetic [16], smooth and pleasant video of the game, as usually seen on TV but without or with minimal human intervention, hence lowering the production cost.

As a computer vision problem, automatic video production of sports events encompasses several open challenges, such as: *where to look at in a scene*? or *what action is taking place*?. Considering a basketball game, for the former problem it implies detecting the ball as it is the more prominent landmark indicating the actionness in the court but given its tendency to be occluded, detecting players to infer the ball position is also necessary. For the latter problem, inferring the state of the game, as in detecting timeouts or defensive-offensive transitions, is of paramount importance to obtain aesthetic productions, as a regular viewer will expect a particular camera framing for a given game state [27].

Figure 1: Given an input video (e.g., 4K) from a central point of view that observes the full basketball court, the proposed system aims to produce an automated virtual camera that tracks the action of interest. The resulting video output (e.g., 720p) is a smooth and pleasant broadcast of the game. A video example can be watched at: https://www.youtube.com/watch?v=3GUsAROG9Z4

In this paper we report solutions to both of the aforementioned problems in the context of an automatic video production system for basketball games. Our proposed system can be cast as a virtual camera system [10], as the camera does not move and it acquires high level resolution images to produce the output video, as it is shown in Fig. 1. This type of systems are also known by the name of video retargeting [30], virtual videography [20], virtual robot camera [22] or re-cinematography [18]. We will review next these terms and related work to this paper.

1.1. Background

The process of creating high quality broadcast production of live sports events is one of high monetary cost, as it requires a coordinated crew of experienced skilled individual with fast response in a changing situation [27]. Some niche sports events cannot cover the high cost of having a

*All authors have contributed equally.

human production crew and hence the following problem arise: *How can a sport event be produced with no human intervention but having an aesthetic quality?*

This problem does not only apply to sports [26] but to any live event, such as a university lecture [38], a news show [36] or a theater play [30], and the proposed solutions have different names across related fields. In robotics the name robot camera was coined more than 20 years ago [21], where the Japanese TV company NHK has a long standing track of research in the topic, from football auto-production [35], to news show [36], including also studies of cameraman performance evaluation [22]. In the robotics field the camera physically moves to perform the auto-production and this property is known in the literature as "real camera" [10]. Using a "real camera" implies solving the so-called visual servoing task [8] which involves applying feed-back control techniques. An example of using a proportional-only controller for automatic basketball production is shown in [5].

In the computer graphics and multimedia information system communities, video re-targeting [30], virtual videography [20] or re-cinematography [18] are common terms to refer to the automatic camera system problem. All of them have in common that usually the automatic camera system problem is approached using a re-sampling of a high-resolution video, hence the camera is static. This type of cameras are known as "virtual cameras" and in sports events it requires to cover the whole pitch or court with one or multiple overlapping cameras. Also all of the aforementioned terminology implies that the solution is based on the art of videography [25], in particular following the filming laws for cinematography [4] that aims at producing videos that follow standard filming cinematic conventions, i.e., camera motion has continuity in velocity and direction aiming at aesthetic video production [17, 18].

Besides the possible categorization of automatic camera system according to its properties, e.g., real-cameras vs virtual cameras or surveillance vs sports events, there are three problems that need to be solved for any automatic camera system [10]:

Camera planning. Relates to the problem "*where should cameras look at*?" and in sports events there is some prior knowledge to solve it, e.g, in basketball important actions usually take place near to the ball. Typical solutions to this problem rely on object detection [23, 28] and some kind of activity recognition [3, 2] based on players when the ball is occluded. It is important to point out that an appropriate solution to this problem should report where to point the central point of the camera as well as the appropriate field of view coverage. Solutions to this problem can involve, among others, mild human intervention as in [15] or [14] or learning directly from data the desired state of the camera as reported in [11] and [12].

Camera control. Relates to the problem "*how should cameras move*?" and its solution differs if the camera moves physically or virtually. In any case, it involves moving the camera from its current configuration to a desired one taking into account where it should look in a scene. Nowadays with the availability of 8K or 4K imaging sensors and better optics it is possible to use the virtual camera approach in sports events but traditionally the use of PTZ cameras [5] or over-rail cameras have been the norm [35].

Camera selection. Relates to the problem "*which camera should be selected*?" and applies in a setup with multiples cameras. Solutions to this problem in sports events are challenging because there can be fast transitions between cameras, and solutions usually need to be based on the type of action happening in the court or pitch. Many solutions are data-driven based on machine learning techniques as reported in [9, 13] for basketball or [37] for football.

1.2. Paper Contributions

In the context of basketball automatic video production, in this paper we report solutions to the aforementioned problems by (*i*) computing the actionness in the court by Gaussian modelling players positions plus a ball awareness rule, and (*ii*) game state recognition via a deep learning architecture using a convolutional neural network based on occupancy maps. We also report how they integrate into a basketball automatic video production system.

Finally, we will release with the paper a multipurpose dataset consisting of (*i*) single camera 4K resolution and (*ii*) twelve-camera 1280×1024 resolution videos of basketball games, available at `https://gsbasketball.github.io`. All video footage have ball, players and referees labels per frame. This dataset was used to develop and test the methods reported in this paper.

1.3. Paper Structure

We start by defining the building blocks of the proposed automatic video production system for basketball and providing an overview of each of them in Section 2. Next, in Section 3, we detail our solution to the problem *where to look at in an scene*? in the context of auto-production of a basketball game. Section 4 reports a method for inferring the game state of a basketball game to help the camera framing process of an auto-production system. Then, in Section 5, we present results of the proposed methods and of their integration for automatic video production system for basketball. Finally, we present our concluding remarks in Section 6.

2. Automatic Video Production System

Given a central camera that is able to cover the full court in a basketball game, our system generates a virtual camera

that tracks all the action of interest, producing as a result a smooth and pleasant broadcast of the game.

In order to tackle the automatic production problem for basketball events, we design a system composed of five components: (*i*) court modeling, (*ii*) player and ball detection, (*iii*) actionness, (*iv*) game state recognition and (*v*) video producer, as it is shown in Fig. 2. These components work together to capture and model 3D geometric and semantic information, aiming at producing an *as seen on TV* automatic broadcast of the game.

2.1. Court Modeling

We consider a single fixed central camera, which is able to cover the full court and remains still while the production. The court modeling component is in charge of estimating the camera pose w.r.t. the court and computing useful geometric guidelines for further components.

Knowing the FIBA, NCAA and NBA court markings, and given an image of the basketball court installation, the court modeling estimates a homography H that maps a known set of at least four (co-planar) 3D points $\{(X_i, Y_i, Z_i)\}$ on the court to its corresponding set of projections $\{(x_i, y_i)\}$ in the image. The world coordinate system origin is at center court, with the XY-plane corresponding to the ground plane. The estimated homography corresponds to a projective camera when applied to ground points, so that camera parameters, K, R and C, can be found from $\mathrm{H}(x, y, 1)^T = \mathrm{KR}\,[\mathrm{I}|{-}\mathrm{C}]\,(X, Y, 0, 1)^T$ [19], where K is the intrinsic matrix, R is a rotation matrix from world to camera coordinates, and C is the camera's center of projection in the world coordinates.

The resulting court model is enriched by computing, among others, the court region of interest in the image and the projection of the court volume, as it is illustrated in Fig. 3. Moreover, from this court volume and its projection, we compute a set of optimal framings by cropping the input image space while keeping a 16:9 aspect ratio. In the simpler case, this set is composed by three framings: a *zoom-out* that covers the full court volume projection, and *left-side* and *right-side* framings, which enclose the left and right half-court volumes projections, respectively. Additional framings can be defined by including crops that enclose specific parts of the court volume, e.g., right-aligned 75% of the left half-court volume.

2.2. Player and Ball Detection

We use an object detector as responsible for detecting instances of semantic objects of three classes: ball, player, and referee. Looking for a reasonable trade-off between running time and accuracy, we selected and slightly modified YoloV3 [32] for this task. The input image is resized to 608×608, and we disconnect layer 36 in the combined feature map and connect instead the layer 11, to be able to detect small objects (up to 16×16) at the resized resolution. Starting from the original weights trained in COCO [24], we train using CrowdHuman dataset [33] and a Google sports dataset [31], where only the class player was learned. Later we include the class ball and perform a fine-tuning procedure in the datasets APIDIS and SPIROUDOME, and finally the third class referee is added and the network is fine-tuned in our twelve-view cameras basketball dataset.

In order to improve the detection accuracy for our single camera problem, we exploit the court modeling in a last fine-tuning stage. For this purpose, all considered images in this process should have a court model that provides a rectangular region containing the court. We crop training images to fit the court region and resize them to 1024×192, which we found respect the aspect ratio from this point of view. The anchors are re-computed at this stage using k-means. Resulting players and ball detections will feed both Actionness and Game state recognition components.

2.3. Actionness

We define *Actionness* as the two-dimensional function, with domain the court plane, that estimates the amount of action that can take place at every location of the basketball court. For that, we model the distribution of detected players in the court and compute some statistics from this distribution. We also include a ball awareness rule to deal with specific situations of the basketball game. Sec. 3 describes the Actionness function in detail and how the main action coordinates are estimated.

2.4. Game State Recognition

Rules of the game production should vary depending on the game state, e.g, a half-court possession could be properly covered by an almost steady camera, whereas a fast-break requires a fast moving camera to not lose the main action. With that in mind, we designed a deep-learning based solution to recognize and localize three game states that will guide the final production of the basketball game, as it is described in Sec. 4.

2.5. Video Producer

The last component of our system is the actual video producer of the game. It receives as inputs the current full court image, a game state and the estimated action coordinates. As output it delivers a virtual camera that optimally frames the main action by cropping the input image and finally resizes it to the output resolution, e.g., 720p or 1080p. The producer also should ensure a smooth as possible production, so following the results of [30], we define a maximum displacement in pixels per frame for the virtual camera.

In our system we consider three game states: half-court, transition and not-playing. The production rules vary depending of the estimated state as it is described below.

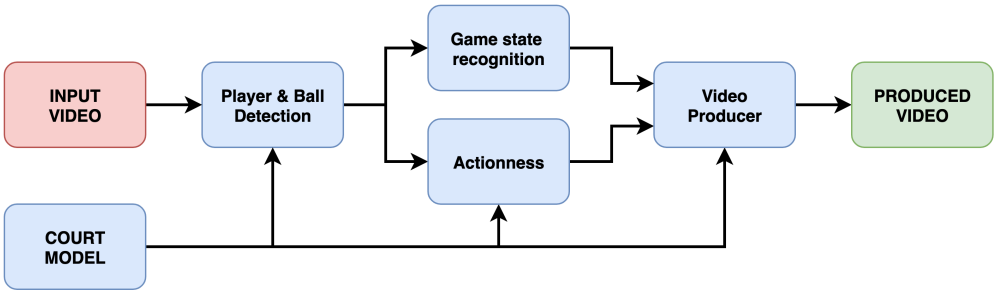


Figure 2: Diagram of the five components of the system.

Figure 3: Court modeling. The projection of a four-meters high volume of the court is drawn in blue, the modeled court markings are presented in orange and magenta.

2.5.1 Transition Production

During a transition, players are moving from one side to the other at different velocities and the camera should be attentive to track the main action. Hence, during a transition state the camera will be almost free, tracking the action.

The production at this state is done by firstly filtering the temporal series of each component of the action coordinates, using the one-euro filter [7] to prevent undesired high-frequency variations. Then a proportional control is used to move the current location of the virtual camera to the filtered action value, without exceeding the maximum displacement. We only take into account the horizontal of the action, so the motion of the virtual camera corresponds to a panning. Finally, given the output of the controller, the input image is cropped centered at this value at a scale computed as the average of the half-court framings.

2.5.2 Half-court Production

During a half-court state the camera will be almost steady covering the corresponding half-court, which is indicated by the game-state recognition state. In the case of a single framing per half-court, once this state is detected the framing of the virtual camera will smoothly move towards the desired framing, both in location and scale. It is important to note that framings could differ in size since they are computed to cover a specific volume projection in image, so a change in scale could be required. If more than one framing is defined per half-court, the virtual camera will move towards the framing closer to the main action coordinates.

2.5.3 Not-playing Production

Not-playing state is formed by the pre-game interval, warm-ups, end of period intermissions and timeouts. At this state there is a single rule of production that is to reach the zoom-out framing and remain there until the state changes. Once the not-playing state is detected, the producer will move the virtual camera from its current position to the zoom-out framing using a linear interpolation between these two framings without exceeding the displacement limit.

3. Actionness

Actionness is defined as the two-dimensional function, with domain the court plane, that estimates the amount of action that can take place at every location of the basketball court. For this purpose, we model the distribution of detected players by projecting their location into the court plane. Then we perform a smoothness procedure to aggregate the spatial information across the court. Finally, the peak location of this distribution is obtained and corrected to ensure that the ball is contained. The resulted value, called *action value*, is used to perform the final video production, as it is described in Sec. 2.5.

3.1. Actionness Function

Assuming that every player is touching the basketball court with their feet, we estimate its location in the image domain as the middle point at the bottom of the detected box. Let $\mathbf{x}_{tl} = (x_{tl}, y_{tl})^T$ and $\mathbf{x}_{br} = (x_{br}, y_{br})^T$ be the top-left and bottom-right corners of a detection, respectively, the player location in the image is given by the middle at the bottom as $\mathbf{x}_c = (x_c, y_c)^T = (0.5 * (x_{tl} + x_{br}), y_{br})^T$. The court location (at zero height) can be found using the Homography computed in the court modeling:

$$\overline{\mathbf{X}}_c = \mathrm{H}\,(x_c, y_c, 1)^T = \left(\overline{X}_c, \overline{Y}_c, \overline{Z}_c\right)^T, \qquad (1)$$

from where the location in court units is given by $\mathbf{X}_c = (X_c, Y_c)^T = \left(\overline{X}_c/\overline{Z}_c, \overline{Y}_c/\overline{Z}_c\right)^T$. Once players are localized over the court, we model each of them as a two-dimensional Gaussian centered at $\mathbf{X}_c$ and diagonal covariance matrix $\mathbf{\Sigma} = \sigma^2 I$, with $\sigma = 0.25\mathrm{cm}$. Accordingly, a set of detections $\{\mathbf{x}^i_{tl}, \mathbf{x}^i_{br}\}$ localized in the court plane at $\{\mathbf{X}^i_c\}$, generate the actionness function given by:

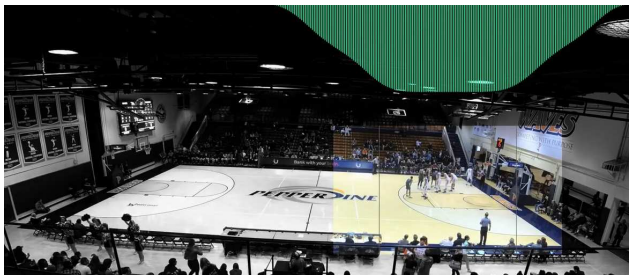
$$\mathscr{A}(\mathbf{X}) = \frac{1}{2\pi\sigma} \sum_i \exp\left(-\frac{1}{2}(\mathbf{X} - \mathbf{X}^i_c)^T \mathbf{\Sigma}^{-1} (\mathbf{X} - \mathbf{X}^i_c)\right).$$

3.2. Action Localization

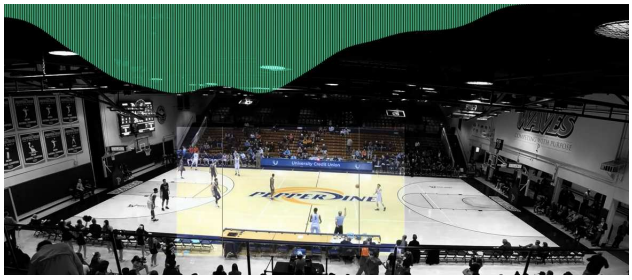
In order to localize the main action of the game, we combine the peak of a smoothed version of the actionness function with a ball awareness rule. The idea behind this combination is twofold: viewers are very interested in the interaction between players and having enough context of what is going on, so trying to frame most of them will likely contain the most interesting actions and provide enough information; on the other hand, the ball should be always contained when framing the most interesting events of the game, e.g., basket, pass, shot, etc.

Since the actionness function is sparse by definition, we produce a smoothed version of the distribution looking for a few number of modes that can make the action localization easier. For this, we locally aggregate the actionness function applying a bi-dimensional convolution with a rectangular window of width and height corresponding to a one quart of court dimensions $\mathscr{A}_s(\mathbf{X}) = \mathscr{A}(\mathbf{X}) \circledast \mathbf{rect}(\mathrm{court}/4)$. The *action value*, noted as $\mathbf{X}^*$, is the location where $\mathscr{A}_s(\mathbf{X})$ reaches its maximum, i.e., $\mathbf{X}^* = \arg\max \mathscr{A}_s(\mathbf{X})$.

In the case when all players are close together, $\mathscr{A}_s(\mathbf{X})$ follows an unimodal distribution where a framing centered at this value will enclose all (or most) players as it is shown in Fig. 4a. Conversely, when players are scattered through the court, function $\mathscr{A}_s(\mathbf{X})$ will have more than one mode and finding the optimal framing is not straightforward. To solve ambiguity and ensure the ball is enclosed, we add a ball awareness rule to correct the action value if the ball is out of the framing. Inspired by the *Rule of Thirds* [16], we use a rule of fifths. The desired framing is centered at $\mathbf{X}^*$ and virtually divided into *five* equally spaced vertical segments. The ball is expected to be located somewhere in between second and fourth segments, i.e., in the inner segments of the framing. If it is not the case, the horizontal of $\mathbf{X}^*$ is corrected until the ball is contained, as it is presented in Fig. 4b. Otherwise, the action value remains unchanged and it is directly passed to the video producer.



(a)



(b)

Figure 4: Actionness: horizontal projection of the Actionness function in green and corresponding framing given by the action value in color. (a) Unimodal distribution generated by players close together. (b) Action value is modified to the right to produce a framing containing the ball inside the inner segments.

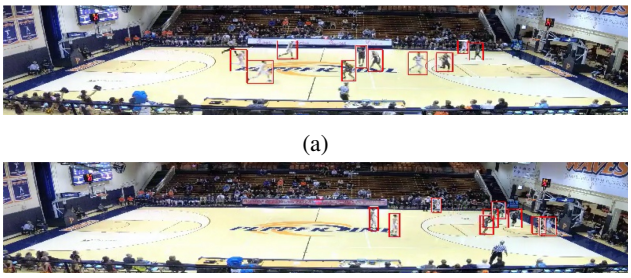
4. Game State Recognition

In the scope of this paper, we will focus in recognition and temporal-detection of game states, more specifically, we will describe a basketball game as a sequence of states, where each state is characterized by a specific distribution of players positions and will determine a specific case for the automatic production.

In brief, the problem of state-of-game analysis is divided on two different axes, state recognition and localization. State recognition, which is the problem of assigning a video to a set of predefined state classes, and state localization, defined as identification of the temporal location where a state is taking place. In this work, we are interested on recognizing three different states: transition, half-court and not-playing. Fig. 5 depicts examples of the aforementioned game states. Below we describe a method to recognize game states on basketball games using player occupancy maps and convolutional neural networks.

4.1. Player Occupancy Map

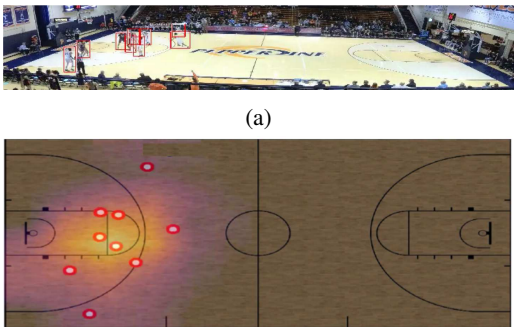
The player occupancy map is based on a $2D$ histogram calculated from the projected bounding boxes provided by a player detector, this kind of method has been used with success in [11, 12] where different histograms resolutions



(a)

(b)

Figure 5: State examples: (a) transition and (b) half-court.



(a)

(b)

Figure 6: Player occupancy map. (a) Example of bounding boxes from our player detector, and (b) projected bounding boxes in the basketball court plane (red dots) overlaid over the player occupancy map ($N_X = 48$, $N_Y = 24$).

were used to determine the pan position for a virtual camera.

We first localize players by projecting their detected boxes in the court plane, as it is presented in Sec. 3.1. Later, to describe the distribution of players we divide the basketball court plane in a $2D$ grid of size (N_X, N_Y). Finally, the occupancy map $\mathbf{X_h}$ is computed counting the number of projected bounding boxes falling within each cell of the grid, as it is illustrated in Fig. 6.

4.2. Recognizing Game States with CNN

Convolutional Neural Networks (CNN) have been widely used for recognizing actions due to their power to encode the information from images [6, 29]. In this paper, we will use a CNN to recognize states of a basketball game using the player occupancy map explained in the preceding Section. Our main assumption resides in identifying the different configurations of players that corresponds to a specific basketball game state using a CNN trained architecture and the player occupancy map. The specificity of the CNN used in our method is illustrated in Fig. 7.

In a nutshell, the input of the network is a $\mathbf{X_h}$ occupancy map of size 24×48, and the neural network architecture is composed of five convolutional layers and two fully connected layers which are in charge of classifying the features from the occupancy map into a specific state class. The number of filters and neurons in the fully connected layers were chosen experimentally with videos that were not included in the training or testing dataset. We have used a Global Average Pooling instead of a simple flattening due to its capacity to capture the information in an aggregated way, without loosing the spatial configuration of the players that we want to keep to identify the states. Finally, to recognize the game states on a video, we apply the trained CNN to the computed $\mathbf{X_h}$ for each frame, and assign the state with the highest output score.

5. Evaluation

In this section, we report a set of experiments to evaluate the performance of the proposed methods in Sec. 3 and Sec. 4. We also report results of the performance of the whole system described in Sec. 2.

5.1. Training and Evaluation Dataset

For training, tuning and testing the proposed methods and the system, we used NCAA basketball video recorded games from men and women. We used a total of 198000 images for training and tuning, i.e., without data augmentation strategies. For testing, we have available 68534 images (≈39.38 min). For each image, we have available human-annotated (*i*) bounding boxes of the categories of interest (player, ball, referee), (*ii*) game states: half-court, transition and not-playing, and (*iii*) production of the game: pan value of the virtual camera and manual crop of input images.

5.2. Evaluation of the Actionness

We firstly evaluate ball and player detection as the actionness is a function of them, and secondly we assess the accuracy error of the resulting framing.

5.2.1 Player and Ball Detection

For evaluating player and ball detection we use the precision-recall curve for each class. We have tested with different values of Intersection over Union (IoU), that cover a range between 0.4 to 0.9, as it is shown in Fig. 8. We present in Table 1, the values of mAP (mean Average Precision) for each variation of IoU. It can be observed that at IoU = 0.5, player detection is very accurate achieving a mAP of $\approx$ 0.9 while ball detection is highly unreliable. Nevertheless, the lack of accuracy in ball detection do not prevent the system for delivering a high quality auto-production.

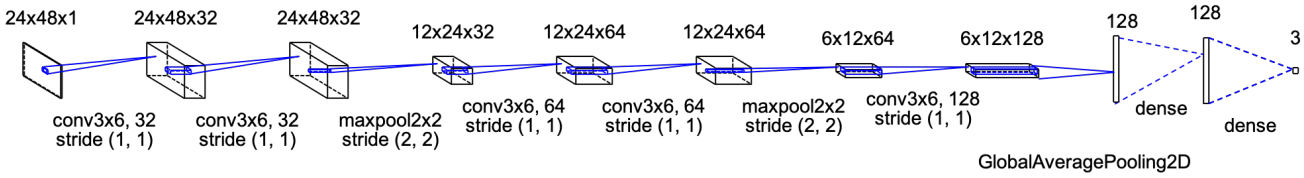
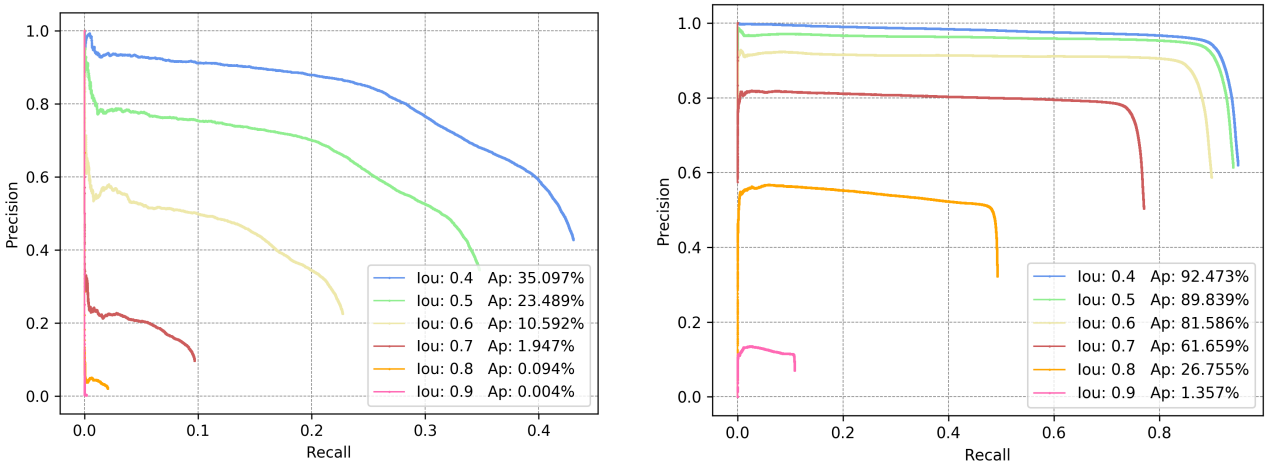


Figure 7: CNN architecture used to recognize states with the player occupancy map.



(a) (b)

Figure 8: Precision-Recall curves for the trained object detector for (a) ball and (b) players

Table 1: mAP metrics ball and player detection

IoU	Ball	Players	mAP
0.4	0.351	0.925	0.638
0.5	0.235	0.898	0.566
0.6	0.106	0.816	0.461
0.7	0.019	0.617	0.318
0.8	0.000	0.268	0.134
0.9	0.000	0.014	0,007

5.2.2 Accuracy Error in Framing

We measure the accuracy error of the resulting production by computing the Mean Absolute Error (MAE) of the action localization $\mathbf{X}^*$ generated by the system (explained in Sec. 3) w.r.t the human-annotated center of the production $\mathbf{X}^{\text{ref}}$:

$$\text{MAE} = \frac{1}{T}\sum_{t=0}^{T-1} |\mathbf{X}_t^* - \mathbf{X}_t^{\text{ref}}| \qquad (2)$$

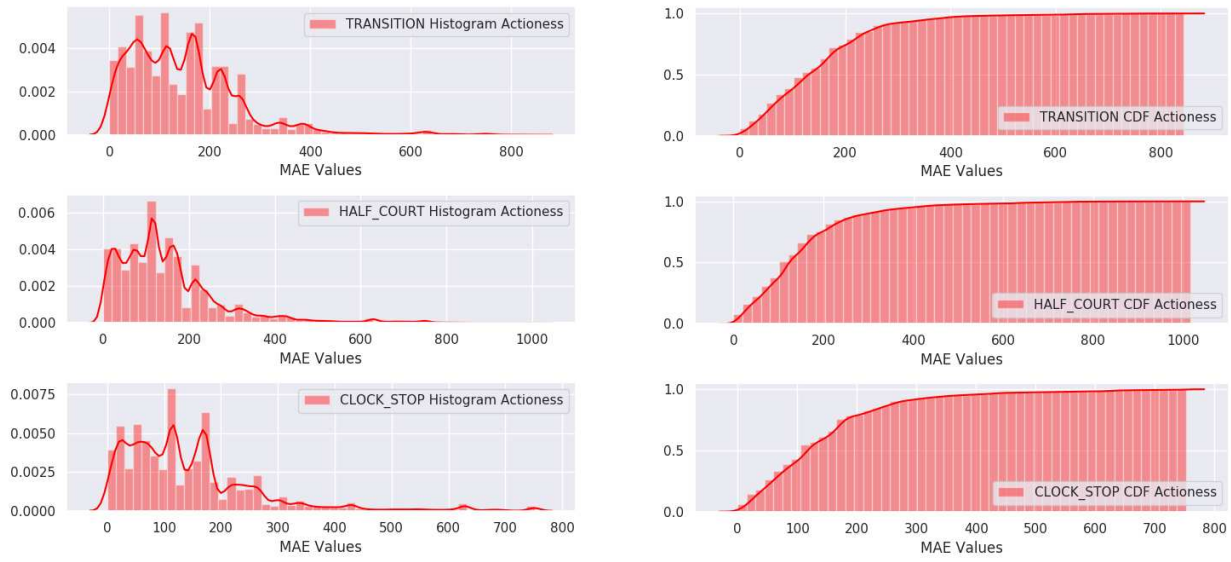
Fig. 9 shows the accuracy error over the testing dataset using a histogram and to provide more insights we compute the CDF from the histogram. This CDF indicates the probability of occurrence of an accuracy error greater than N pixels. We also disaggregate results per game state. It is worth noting that 95th percentile of the CDF is less than 250 pixels, i.e., for an auto-produced video of 720p, the ball is contained in the resulting frame at least 95% of the time.

5.3. Evaluation of State Recognition

To show the performance of the game state recognition, we present a temporal analysis for a sequence of frames in Fig. 10a. It is possible to observe that in a sequence of a basketball game, we can predict the correct state of the game without having large sequences of frames with a mistaken state. In addition, for measuring the capacity of classifying a state, we have used a confusion matrix as it is shown in Fig. 10b. The best performance of the system was obtained in the transition state with 0.91. In the opposite side, the not-playing state got a score of 0.78, having a confusion of 0.16 with the half-court state.

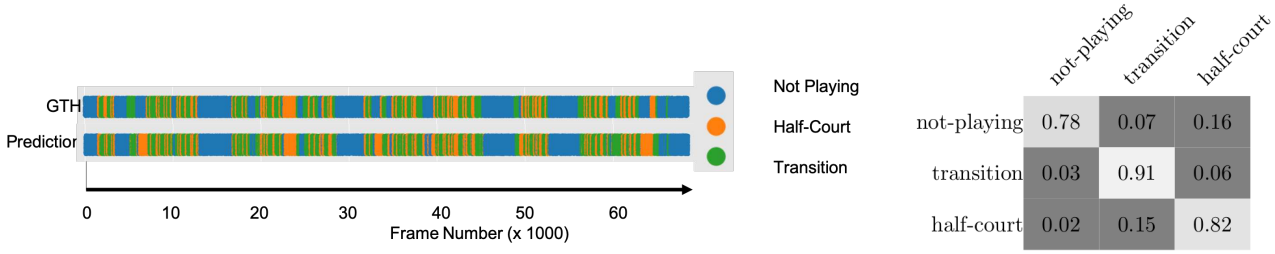
5.4. Qualitative Evaluation of the System

Aiming at assessing the subjective quality of the resulting basketball production, we asked 10 persons to rate from 0 (Absolutely not) to 5 (Completely, yes) the following two question about the quality of 84 auto-produced video clips containing a sequence of a basketball game:



(a) (b)

Figure 9: Actionness evaluation. (a) MAE histogram of the action localization w.r.t. the human-annotated center of the production. (b) CDF of histogram of accuracy error showing the probability of having a MAE less than N pixels.



(a) (b)

Figure 10: Results of the game state recognition presented (a) as a sequence of frames continuously injected into the system and (b) in terms of classification performance expressed as a confusion matrix.

- **Q1**. Is the main action of the game visible clearly?
- **Q2**. Is the ball properly enclosed in the production?

For **Q1** the mean rate was 4.63 with a standard deviation of 0.67 and for **Q2** 4.30 and 0.97 respectively. Overall the comments from evaluators were positive praising the steadiness and on the spot framing while a half-court state. Main opportunities for improvement are in the jittering during transition when the ball is not detected and lagging behind in long passes.

A Youtube video playlist is available with examples of auto-produced video clips of basketball games at https://www.youtube.com/playlist?list=PL03T17ARL5kHvSyMZj3xm8jbfT9aJ3OXf.

6. Conclusion

We have presented a video system able to automatically produce a smooth and pleasant broadcast of basketball games using a single fixed 4K camera. The system automatically detects and localizes players, ball and referees, to recognize main action coordinates and game states yielding to a professional cameraman-like production of the basketball event. Without the need for expensive or additional hardware installation, the system provides a high quality and low latency (less than 10 seconds processing in the cloud) automatic video production of live basketball games and is scalable to indoor and outdoor sporting events. Due to its bottom-up design, 3D geometry awareness and deep learning capabilities, the quality of the auto-production will be constantly enriched and improved as more venues, data, and feedback from fans are available.

References

[1] Unleashing the New Era of Long-Tail OTT Content. https://www.sportbusiness.com/2020/01/genius-sports-whitepaper-unleashing-the-new-era-of-long-tail-ott-content/. Accessed: 2020-03-02.

[2] Yasuo Ariki, Shintaro Kubota, and Masahito Kumano. Automatic production system of soccer sports video by digital camera work based on situation recognition. In *Eighth IEEE International Symposium on Multimedia (ISM'06)*, pages 851–860. IEEE, 2006.

[3] Alina Bialkowski, Patrick Lucey, Peter Carr, Simon Denman, Iain Matthews, and Sridha Sridharan. Recognising team activities from noisy data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 984–990, 2013.

[4] John Cantine et al. *Shot by shot: A practical guide to filmmaking*. ERIC, 1995.

[5] Peter Carr, Michael Mistry, and Iain Matthews. Hybrid robotic/virtual pan-tilt-zom cameras for autonomous event recording. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 193–202, 2013.

[6] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733, July 2017.

[7] Géry Casiez, Nicolas Roussel, and Daniel Vogel. One-euro filter: a simple speed-based low-pass filter for noisy input in interactive systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2527–2530, 2012.

[8] François Chaumette, Seth Hutchinson, and Peter Corke. *Visual Servoing*, chapter 34, pages 841–866. Springer, 2016.

[9] Fan Chen and Christophe De Vleeschouwer. Personalized production of basketball videos from multi-sensored data under limited display resolution. *Computer Vision and Image Understanding*, 114(6):667–680, 2010.

[10] Jianhui Chen and Peter Carr. Autonomous camera systems: A survey. In *AAAI Workshops*, 2014.

[11] Jianhui Chen and Peter Carr. Mimicking human camera operators. In *2015 IEEE Winter Conference on Applications of Computer Vision*, pages 215–222. IEEE, 2015.

[12] Jianhui Chen, Hoang M Le, Peter Carr, Yisong Yue, and James J Little. Learning online smooth predictors for realtime camera planning using recurrent decision trees. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4688–4696, 2016.

[13] Fahad Daniyal and Andrea Cavallaro. Multi-camera scheduling for video production. In *2011 Conference for Visual Media Production*, pages 11–20. IEEE, 2011.

[14] Eric Foote, Peter Carr, Patrick Lucey, Yaser Sheikh, and Iain Matthews. One-man-band: A touch screen interface for producing live multi-camera sports broadcasts. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 163–172, 2013.

[15] Vamsidhar Reddy Gaddam, Ragnhild Eg, Ragnar Langseth, Carsten Griwodz, and Pål Halvorsen. The cameraman operating my virtual camera is artificial: Can the machine be as good as a human? *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 11(4):1–20, 2015.

[16] Raghudeep Gadde and Kamalakar Karlapalem. Aesthetic guideline driven photography by robots. In *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

[17] Michael L Gleicher and Feng Liu. Re-cinematography: improving the camera dynamics of casual video. In *Proceedings of the 15th ACM international conference on Multimedia*, pages 27–36, 2007.

[18] Michael L Gleicher and Feng Liu. Re-cinematography: Improving the camerawork of casual video. *ACM transactions on multimedia computing, communications, and applications (TOMM)*, 5(1):1–28, 2008.

[19] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, USA, 2 edition, 2003.

[20] Rachel Heck, Michael Wallick, and Michael Gleicher. Virtual videography. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 3(1):4–es, 2007.

[21] Akio Ishikawa, Daiichiro Kato, and Hiroshi Fukushima. Measurement method of zooming by a cameraman. In Stephen K. Park and Richard D. Juday, editors, *Visual Information Processing VII*, volume 3387, pages 13 – 24. International Society for Optics and Photonics, SPIE, 1998.

[22] Daiichiro Kato, Akio Ishikawa, Takao Tsuda, Shigeru Shimoda, and Hiroshi Fukushima. Automatic control of a robot camera for broadcasting and subjective evaluation and analysis of reproduced images. In *Human Vision and Electronic Imaging V*, volume 3959, pages 468–479. International Society for Optics and Photonics, 2000.

[23] Masahito Kumano, Yasuo Ariki, and Kiyoshi Tsukada. A method of digital camera work focused on players and a ball. In *Pacific-Rim Conference on Multimedia*, pages 466–473. Springer, 2004.

[24] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[25] Eric Margolis and Luc Pauwels. *The Sage handbook of visual research methods*. Sage, 2011.

[26] Kentaro Matsui, Masaki Iwase, Masato Agata, Toshimi Tsu Tanaka, and Noboru Ohnishi. Soccer image sequence computed by a virtual camera. In *Proceedings. 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No. 98CB36231)*, pages 860–865. IEEE, 1998.

[27] Jim Owens. *Television sports production*. CRC Press, 2015.

[28] Hemanth Pidaparthy and James Elder. Keep your eye on the puck: Automatic hockey videography. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1636–1644. IEEE, 2019.

[29] Z. Qiu, T. Yao, and T. Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *2017 IEEE*

International Conference on Computer Vision (ICCV), pages 5534–5542, Oct 2017.

[30] Kranthi Kumar Rachavarapu, Moneish Kumar, Vineet Gandhi, and Ramanathan Subramanian. Watch to edit: Video retargeting using gaze. In *Computer Graphics Forum*, volume 37, pages 205–215. Wiley Online Library, 2018.

[31] Vignesh Ramanathan, Jonathan Huang, Sami Abu-El-Haija, Alexander N. Gorban, Kevin Murphy, and Li Fei-Fei. Detecting events and key actors in multi-person videos. *CoRR*, abs/1511.02917, 2015.

[32] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv*, 2018.

[33] Shuai Shao, Zijian Zhao, Boxun Li, Tete Xiao, Gang Yu, Xiangyu Zhang, and Jian Sun. Crowdhuman: A benchmark for detecting human in a crowd. *CoRR*, abs/1805.00123, 2018.

[34] Graham Thomas, Rikke Gade, Thomas B Moeslund, Peter Carr, and Adrian Hilton. Computer vision for sports: Current applications and research topics. *Computer Vision and Image Understanding*, 159:3–18, 2017.

[35] Takao Tsuda, Daiichiro Kato, Akio Ishikawa, and Seiki Inoue. Automatic tracking sensor camera system. In *Machine Vision Applications in Industrial Inspection IX*, volume 4301, pages 144–153. International Society for Optics and Photonics, 2001.

[36] Takao Tsuda, Makoto Okuda, Kazutoshi Mutou, Hitoshi Yanagisawa, Noriyoshi Kubo, and Yoshikatsu Date. A high-accuracy image composition system using a mobile robotic camera. *SMPTE motion imaging journal*, 117(2):30–37, 2008.

[37] Jinjun Wang, Changsheng Xu, Engsiong Chng, Hanqing Lu, and Qi Tian. Automatic composition of broadcast sports video. *Multimedia Systems*, 14(4):179–193, 2008.

[38] Takao Yokoi and Hironobu Fujiyoshi. Virtual camerawork for generating lecture video from high resolution images. In *2005 IEEE International Conference on Multimedia and Expo*, pages 4–pp. IEEE, 2005.