

Tracking Basketball Shots – Preliminary Results

Juozas Širmenis^a, Mantas Lukoševičius^a

^aFaculty of Informatics, Kaunas University of Technology, Kaunas, Lithuania

Abstract

In this work we present initial research results for creating a system that tracks basketball practice shots from a single camera. We investigate different background subtraction and ball recognition methods, report their accuracy on a basketball video with a static position of the camera, single shooter, ball, and board visible. This preliminary research will allow us to implement better machine-learning-based solutions, as well as annotate data for their training.

Keywords

Single-camera, Basketball video analysis, Background subtraction, Object tracking, Ball recognition, Computer vision

1. Introduction

Technological growth leads to more complex devices which help the sport industry to create more quality and enjoyable content. Some devices are attached directly to the athletes or their wear to track motions, progress, and analyze techniques [1]. Algorithms are also used for real-time tracking in a sports game to capture, analyze and present data graphically [2].

Basketball is the most popular sport in Lithuania and one of the most popular in the world. Players interact with each other and with a ball to create a spectacular event. Object tracking is being used in sports to track players or/and balls. The collected data is used to provide detailed player information, progress, analyze movement, tactics and strategy next steps in other matches. Based on experts, the best way to improve individually is to learn by repeatable practice [3] and receive proper feedback [4].

In this paper, we want to reach a solution that would help basketball players to improve their performance by repeatable practice and instant feedback. Individual training frequently means shooting alone. Usually, players do not have large amounts of equipment, so we will try to create a solution with a single mobile camera. Along with complicated algorithms, the system will recognize and track the basketball. Collected data will be analyzed and presented with the count of missed and made shots, practice progress, and advice on how to grow. The athlete will be able to fully focus on making shots and improving their techniques.

IVUS2021: Information Society and University Studies 2021, April 23, 2021, Kaunas, Lithuania

✉ juozas.sirmenis@ktu.edu (J. Širmenis); mantas.lukosevicius@ktu.edu (M. Lukoševičius)

🌐 <https://mantas.info/> (M. Lukoševičius)

🆔 0000-0001-7963-285X (M. Lukoševičius)

© 2021 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).



CEUR Workshop Proceedings (CEUR-WS.org)

2. Related Work

Some products already do have similar functionality. In this section, we take a deeper look into already existing relevant solutions, and their used ball and movement recognition methods.

2.1. Commercial Products

There are several similar alternatives on the market. *The Shot Tracker* founded by Bruce Ianni and Davyeon Ross is using three different components: anchors, ball, and player sensors to real-time analyzes of shots and players position [5]. *Wilson X Connected Basketball* designed and created by *Wilson LABS* with additional help of *Wilson Sporting Goods* is based on a sensor inserted in the ball and the installed app connected via *Bluetooth* where all the results about the shots are shown [6]. *HomeCourt* launched in 2018 for iOS devices by *NEX Team* and co-founder David Lee is based on artificial intelligence (AI) technology and computer vision to track and analyze shots, ball, and player movement, and give them proper real-time feedback [7, 8].

2.2. Main Stages of Ball Recognition

There are a lot of different ways to recognize and track objects in video. Firstly, the ball has to be separated and recognized from other objects. Secondly, the movement of the ball has to be noticed and analyzed. In this chapter, we discuss the most popular and efficient ways.

2.2.1. Ball Recognition

To be able to track a ball, first of all, we have to recognize and separate it from the other objects in the video. It is clear that the ball will move and will be as close as possible to a round shape. One of the most popular solutions is to use color segmentation. Often basketball is in one color and because of that, it is easy to recognize it in the video. It is possible to analyze few video files and extract basketball shape histograms manually, that later are used as the main source of possible basketball color range, which enables the recognition of the ball [9].

The color of the ball in the video can change due to external interference. The best solution for this is to convert red-green-blue (RGB) color space to hue-saturation-value (HSV) space. In the sequel ball's color threshold is determined and after dilation and erosion ball is found [10]. Sometimes video is extracted to HSV color space and then used coarse strategy for not-ball candidates elimination because of their color and shape. Finally, the most likely ball is found based on a result from connecting ball's color and circularity analyzes [11].

The shape recognition of the ball is based on eccentricity. It is calculated as the ratio of the height to the width of the minimum rectangle which has to contain all the points of the shape. After that, some non-ball objects may be determined as balls. To avoid that object has to be analyzed based on its circularity which can be defined of the shape's area and perimeter [12].

Sometimes a shape filter is combined with a size filter. Due to different video conditions, one source states that the size of the ball should be approximately between 3 to 15 pixels [13], other one 5 to 30 pixels [14], but based on the analysis of our test data, the ball's size should be approximately 15 to 40 pixels. To achieve more accurate results in the ball's recognition, along with size and shape filters, a compactness filter is added too [12, 14, 15]. Compactness is the

object characteristic that describes how close a shape is to a circle. If compactness gets a result of 1, the shape is perfectly round and may be called a circle. Compactness is calculated as

$$C = \frac{P^2}{4\pi A}, \quad (1)$$

where P is the perimeter and A is the area of the shape [16].

During the research, we analyzed: color segmentation, coarse strategy, size, shape and compactness filters, and other ball recognition techniques. Also, we noticed that other similar systems did not use just one algorithm, usually, it takes 2 or more to recognize the ball from the background. We also will experiment with different combinations of algorithms and methods.

2.2.2. Movement Recognition

The most popular methods for a moving ball recognition are: frame difference [9, 15] and background subtraction [13], or the the combination of the two [17, 18].

Frame difference is the method where the absolute difference is calculated between two or three consecutive frames. The difference between the three frames is calculated by finding the absolute difference between the previous and current frames, and the same with the current and the next one. Subsequently, both results are logically added to each other. If their position of pixels is the same in both the difference frames images, these are the objects which moved in taken frames [18].

The background subtraction method creates a reference background image and compares each video frame with it to separate foreground from background [18]. Moving objects are indicated by pixels that stand out drastically compared to the background model [19].

To simplify calculations, the frame difference method is used in the background subtraction method along with a combination of the morphological operation and edge detection [12, 14]. Morphological operation fills the empty gaps between the segmented regions and removes the noise [12]. Edge detection is used to define moving objects and generate ball candidates [14].

3. Preliminary Methods and Results

Here we describe our preliminary methods and results. To properly recognize and track a ball, we go through many different stages. First of all, we separate the ball from the background. Then analyze its size and circularity to make sure it is a correct ball and then validate its movement. After all, we implement a couple of filters to improve the accuracy.

3.1. Video Sequences Used

We do our preliminary research on several self-recorded videos, with different angle of a static camera, background lighting, and the color of the player's clothing. Footage contains a single player and ball with a well-seen basketball board. Video is MP4 format, 1920x1080 resolution, and 30 frames per second. For this preliminary analysis, a single five second video sequence with the ball always visible was selected.

3.2. Background Subtraction

First of all, we separate foreground and background. For that, we use a background subtraction. For simplicity, we are going to use the OpenCV library, because almost all of the needed methods are implemented there already. We will experiment with 3 different background subtraction methods from the OpenCV library: Mixture of Gaussians (MOG), Mixture of Gaussians 2 (MOG2), and K-nearest neighbors (k -NN) [20].

MOG uses a method to calculate a mixture of k Gaussian distributions, where k is between 3 and 5, for each pixel in that frame. Foreground and background colors are expressed by different distributions. The weight of each distribution represents the duration of how long a pixel's color stayed in one position. So, if a color in that pixels stays for a longer amount and is more static, the weight of distribution is high – that may be classified as background [21].

MOG2 is also a mixture of Gaussians but more efficient and has additional features. MOG method had a limited amount of used distributions (3 to 5), MOG2 solves that problem. It adds more adaptivity and flexibility due to the dynamic number of distributions. In addition, MOG2 implemented functionality of shadow detection [22].

The main idea of the k -NN method is that k -Nearest-Neighbors-based recursive equations automatically recalculates values of parameters for the Gaussian mixture model and at the same time, for each pixel, picks the appropriate amount of components. This method helps to improve kernel density estimation [23].

Both methods MOG2 and k -NN have an opportunity to subtract a background with a shadow detection. We will experiment with both of these methods.

3.3. Ball Recognition

After the background subtraction, we are going to use either morphological operations (open, close, dilate) [24] or blurring method [25] to reduce noise and fill small gaps inside the moving objects. Subsequently, we will add the Canny edge detection method [26], and finally, then all this is done, we will try to find all the circles in the processed video frames with the Hough Circles method [27] which also already exists in the OpenCV library.

To check which combination of methods gives the best results, we calculate the percentage of frames in which a ball was detected. We will check if a ball candidate occurs once in one frame. If there are none or more than one, it means that the ball recognition did not work.

Table 1

First accuracy results (in percent)

Method	GaussianBlur	MedianBlur	Morphology
MOG	30	38	29
MOG2 (on)	1	2	39
k -NN (on)	0	0	4
MOG2 (off)	0	1	38
k -NN (off)	0	1	14

Table 1 contains information about accuracy results. In rows, there are different background

subtraction methods we used, and in columns, there are different methods for reducing noise and filling holes. Near MOG2 and k -NN methods “on” and “off” means if shadow detection was activated.

3.3.1. Centre Filter

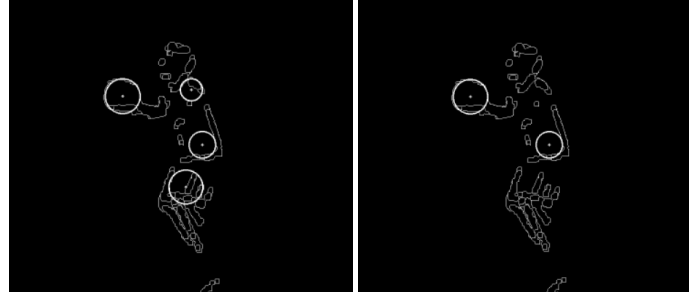


Figure 1: Before and after Centre filter

We noticed that some circles found by Hough Circles method [27] have noise inside them (Figure 1). Usually the real-ball candidate has an empty, black hole inside it after Canny edge detection [26]. So, to increase the accuracy, we took the value of the center of the ball 10 pixels wide to check if there are some white pixels in it. If so, this is not a ball. We call this a “Centre filter”. After we applied it, we recalculated the accuracy results again, as shown in Table 2.

Table 2

Second accuracy results after Centre filter (in percent)

Method	GaussianBlur	MedianBlur	Morphology
MOG	47	49	38
MOG2 (on)	55	56	66
k -NN (on)	38	50	49
MOG2 (off)	33	33	51
k -NN (off)	13	40	32

3.3.2. Motion Filter

By visually analyzing the detected circles in each frame, we can confirm that the best accuracy was observed in MOG2 (with shadow detection) and Morphology method combination. But we also noticed that the real-ball candidates are almost always moving. The speed of his movement is lower than 80 pixels per frame. It means that we can add another filter and call it “Motion filter”, where all the ball candidates will be compared to check if they moved enough to call them a “ball”. We calculate Euclidean distance between the ball-like objects.

The Motion filter works simply and has two layers. In the first one, all possible ball candidate positions in the current frame are compared with the ones in the previous frame, and if they match within 80 pixels, these two positions are possible candidates of a moving ball.

The second layer updates the probability of a ball-like candidate being a real ball. After two ball candidates from different frames passed the position comparison test, they are checked if the previous one in the last frame was also called a possible real ball candidate. If it was, we can call the current ball more confidently the real ball. Our model requires 12 frames at the beginning of the video where the real ball candidate may be designated as a true ball.

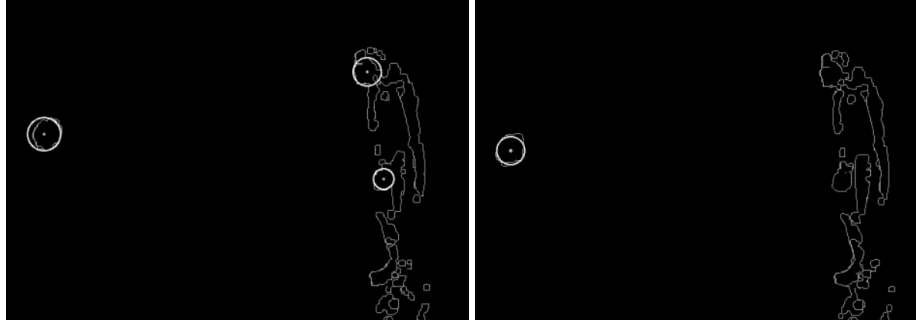


Figure 2: Before and after Motion filter

If there are no ball candidates in the last frame, we are going through a few of the last true-ball positions and count the distance averages between them and the ball-like objects in the current frame. The lowest average is likely to indicate a real-ball candidate. We can see the results after the motion filter is activated in Figure 2.

3.4. Grid Search

The methods and functions we used have many parameters that influence the accuracy results. We used automatic grid search that changes the parameters and records the accuracy. We measured the accuracy on a set of five 3-5 seconds-long self-recorded videos, in the same format as the tested one, with different angles of the static camera, background lighting, and the color of the player's clothing.

The Mixture of Gaussians 2 method reaches the highest accuracy when shadow detection is activated and the history parameter is set to 500, *varThreshold* to 12. For morphological operation, the optimal matrix boundaries of the kernel function are 11, the number of iterations for dilate function is 2. For the best performance of the Canny algorithm, *threshold1* is set to 150 and *threshold2* is set to 300. The best accuracy of the HoughCircles method was observed when *dp* is set to 2, *minDist* to 450, *param1*: 300, *param2*: 21, *minRadius*: 10, *maxRadius*: 50.

3.5. Methods Summary

The block diagram in Figure 3 shows the sequence of methods used. After filters were added, the accuracy of the MOG2 (with shadow detection) and Morphology methods combination increased to 75 %. After the grid search, the average accuracy in the video array was 62.2 %, and the highest individual video - 86.5 %.

During video analyzes, we can confirm that the ball was recognized almost in every well-seen frame. A 100 percent accuracy is not possible because we use 12 frames to initialize the

Motion filter also it is possible that the ball will be hidden behind the shooter's body or get outside the frame of the video.

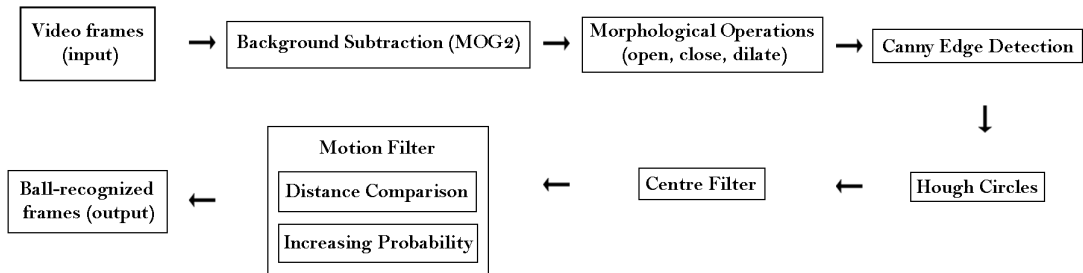


Figure 3: Block diagram for ball recognition

4. Preliminary Results

After we found the best combination of methods and their parameters for ball detection, we implemented a trajectory calculation and prediction for a preliminary shot tracking.

4.1. Trajectory calculation

A free-flying ball has a trajectory of a parabola. We used *polyfit* and *poly1d* methods from *Python* library *NumPy* [28] to find the three polynomial coefficients of the quadratic equation (parabola) and visualize it.

The first 10 consecutive positions of the detected true-ball were selected to find preliminary polynomial coefficients. After that, we draw a predicted trajectory as shown in Figure 4.

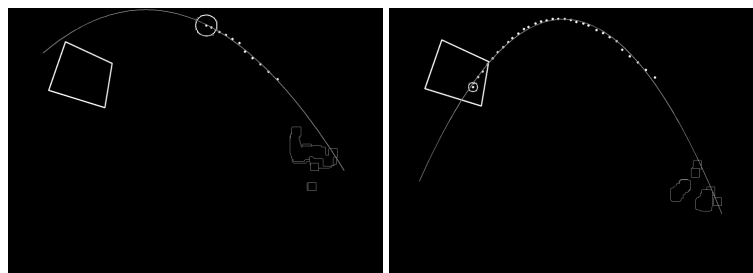


Figure 4: Trajectory calculation and prediction

The thin white line represents the trajectory and wider white dots actual true-ball positions. The recalculate and improve the trajectory prediction after each new basketball is detected.

4.2. Preliminary statistics of shots

Our current system can only count if the ball did not change its trajectory when passing through the hoop and net. Unfortunately, this can mean both a perfect shot or a completely missed one. There are many different ways to make a shot, so our system has to improve here significantly.

5. Discussion and Future Work

Our implemented methods also could be used in other sports events because we used universal solutions. We have to keep in mind that our hard written variables, such as 80 pixels or 12 frames, in the Motion filter can be an obstacle because of the different sizes and speed of the ball, the distance between the camera, and video resolution. If criteria are met as it is in our test file, when the ball may be detected in other sports events.

To confirm this idea, we tested our model with other sport's videos found on Youtube. In volleyball, the ball was recognized despite the camera and people in the background moving a little, and in soccer, a ball was recognized even through the net Figure 5.



Figure 5: Results in volleyball and soccer videos. Top left frame is from a video by Meneo “Tarptautinis moteru papludimio tinklinio turnyras amber cup by kredito garantas 2019” – youtube, <https://youtu.be/Lwj4geVLNwk>, 2020. Bottom left from a video by RDM Football “Raw #3 | knuckleballs, curves, fails and more” – youtube, <https://youtu.be/Vn6epoV7yDk>, 2018.

We want to develop this project to properly track the trajectory of the ball and count shot statistics in a basketball practice using only a mobile phone. For this, the current system has to improve in accuracy and efficiency, include methods that analyze shots, and has to be adapted for mobile devices. Based on the recorded data, the progress of the shooter will be shown along with feedback on the speed, trajectory, and consistency of the shots.

The collected data, tested methods and algorithms will help us employ machine learning in our project. Frames with correct ball recognition will be very helpful as annotated data for training machine learning algorithms. State-of-the-art results in this field are achieved using convolutional neural networks [29, 30, 31]. Our future work is to research the accuracy and computational costs of these methods and incorporate them in our system for benefit.

References

- [1] D. Fitzgerald, J. Foody, D. Kelly, T. Ward, C. Markham, J. McDonald, B. Caulfield, Development of a wearable motion capture suit and virtual reality biofeedback system for the instruction and analysis of sports rehabilitation exercises, in: 2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2007, pp. 4870–4874. doi:10.1109/IEMBS.2007.4353431.
- [2] T. von der Grün, N. Franke, D. Wolf, N. Witt, A. Eidloth, A real-time tracking system for football match and training analysis, in: Microelectronic systems, Springer, 2011, pp. 199–212.
- [3] C. A. Wrisberg, Sport skill instruction for coaches, Human Kinetics, 2007.
- [4] M. Kangalgil, F. Özgül, Use of feedback in physical education and sports lessons for student point of view., Universal Journal of Educational Research 6 (2018) 1235–1242.
- [5] Shottracker | automatically captures statistics for your entire team, <https://shottracker.com/faq>, 2021. (Accessed on 03/07/2021).
- [6] Hands-on test: Wilson x connected basketball | e&t magazine, <https://eandt.theiet.org/content/articles/2017/05/hands-on-test-wilson-x-connected-basketball/>, 2017. (Accessed on 03/07/2021).
- [7] The zoom for basketball: Homecourt makes its product free to use, <https://www.sportstechie.com/nba-basketball-homecourt-nex-team>, 2020. (Accessed on 03/07/2021).
- [8] M. Laukyte, Disruptive technologies and the sport ecosystem: A few ethical questions, Philosophies 5 (2020) 24.
- [9] H.-T. Chen, M.-C. Tien, Y.-W. Chen, W.-J. Tsai, S.-Y. Lee, Physics-based ball tracking and 3d trajectory reconstruction with applications to shooting location estimation in basketball video, Journal of Visual Communication and Image Representation 20 (2009) 204–216.
- [10] H.-I. Lin, Y.-C. Huang, Ball trajectory tracking and prediction for a ping-pong robot, in: 2019 9th International Conference on Information Science and Technology (ICIST), IEEE, 2019, pp. 222–227.
- [11] Xiao-Feng Tong, Han-Qing Lu, Qing-Shan Liu, An effective and fast soccer ball detection and tracking method, in: Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004., volume 4, 2004, pp. 795–798 Vol.4. doi:10.1109/ICPR.2004.1333892.
- [12] B. Chakraborty, S. Meher, A real-time trajectory-based ball detection-and-tracking framework for basketball video, Journal of Optics 42 (2013) 156–170.
- [13] B. Chakraborty, S. Meher, A trajectory-based ball detection and tracking system with applications to shot-type identification in volleyball videos, in: 2012 International Conference on Signal Processing and Communications (SPCOM), IEEE, 2012, pp. 1–5.

- [14] B. Chakraborty, S. Meher, Real-time position estimation and tracking of a basketball, in: 2012 IEEE International Conference on Signal Processing, Computing and Control, IEEE, 2012, pp. 1–6.
- [15] H.-T. Chen, W.-J. Tsai, S.-Y. Lee, J.-Y. Yu, Ball tracking and 3d trajectory approximation with applications to tactics analysis from single-camera volleyball sequences, *Multimedia Tools and Applications* 60 (2012) 641–667.
- [16] R. S. Montero, E. Bribiesca, State of the art of compactness and circularity measures, in: *International mathematical forum*, volume 4, 2009, pp. 1305–1335.
- [17] B. Chakraborty, S. Meher, A trajectory-based ball detection and tracking system with applications to shooting angle and velocity estimation in basketball videos, in: 2013 Annual IEEE India Conference (INDICON), IEEE, 2013, pp. 1–6.
- [18] E. Schmidt, Measuring the speed of a floorball shot using trajectory detection and distance estimation with a smartphone camera: Using opencv and computer vision on an iphone to detect the speed of a floorball shot, 2016.
- [19] S.-C. S. Cheung, C. Kamath, Robust background subtraction with foreground validation for urban traffic video, *EURASIP Journal on Advances in Signal Processing* 2005 (2005) 1–11.
- [20] Opencv: cv::backgroundsubtractor class reference, https://docs.opencv.org/3.4/d7/df6/classcv_1_1BackgroundSubtractor.html, 2021. (Accessed on 03/18/2021).
- [21] L. Marcomini, A. Cunha, A comparison between background modelling methods for vehicle segmentation in highway traffic videos, *arXiv preprint arXiv:1810.02835* (2018).
- [22] A. Mordvintsev, A. K. Revision, Background subtraction — opencv-python tutorials 1 documentation, https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_video/py_bg_subtraction/py_bg_subtraction.html, 2013. (Accessed on 03/20/2021).
- [23] Z. Zivkovic, F. Van Der Heijden, Efficient adaptive density estimation per image pixel for the task of background subtraction, *Pattern recognition letters* 27 (2006) 773–780.
- [24] Opencv: Morphological transformations, https://docs.opencv.org/master/d9/d61/tutorial_py_morphological_ops.html, 2021. (Accessed on 03/22/2021).
- [25] Opencv: Smoothing images, https://docs.opencv.org/master/d4/d13/tutorial_py_filtering.html, 2021. (Accessed on 03/22/2021).
- [26] Opencv: Canny edge detection, https://docs.opencv.org/master/da/d22/tutorial_py_canny.html, 2021. (Accessed on 03/21/2021).
- [27] Opencv: Hough circle transform, https://docs.opencv.org/3.4/d4/d70/tutorial_hough_circle.html, 2021. (Accessed on 03/21/2021).
- [28] numpy.polyfit — numpy v1.20 manual, <https://numpy.org/doc/stable/reference/generated/numpy.polyfit.html>, 2020. (Accessed on 05/10/2021).
- [29] K. Kang, H. Li, T. Xiao, W. Ouyang, J. Yan, X. Liu, X. Wang, Object detection in videos with tubelet proposal networks, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 727–735.
- [30] Z.-Q. Zhao, P. Zheng, S.-t. Xu, X. Wu, Object detection with deep learning: A review, *IEEE transactions on neural networks and learning systems* 30 (2019) 3212–3232.
- [31] H. Zhu, H. Wei, B. Li, X. Yuan, N. Kehtarnavaz, A review of video object detection: Datasets, metrics and methods, *Applied Sciences* 10 (2020) 7834.