NBA Game Replay Auto-Editor

Chen Wang 101052330

Pengliang Zhang 101014341

April 18, 2020

Abstract

Our project provides the user a way to view basketball match videos with special effects applied in real-time. We developed an algorithm to detect momentum shifts within the court by tracking the court borders, players, and the location of the basketball. Overall, our project is able to detect court borders, players, the location of the basketball and apply special effects on certain areas of the video based on analysis in real time.

1 Introduction

The project consists of 5 parts, court detection, player detection, ball detection, applying special effects on certain areas, and the algorithm for analyzing momentum shifts on the court.

1) Court detection – Locating the upper, lower, left, and right borders by using a Canny edge detector and Hough transform.

2) Player detection – Locating individuals in the videos using histogram of oriented gradients, color block detection and single shot detector.

3) Ball detection – Detecting circle objects by using blurring and Hough transform.

4) Applying special effects – Applying convolution with different filter kernels on certain areas of the image.

5) Analyzing momentum shifts on the court – Analyzing results from detection to mark the correct objects and provide areas for applying special effects.

The data used for this project is acquired from the NBA official website for game replays. The replays are then edited to remove advertisements and non-game play moments. During the replay, the camera angle is changing, zooming in and out constantly to track the action on the court, which significantly increases the difficulty level for detection. We specifically selected replays that are recorded by one camera only to avoid further increasing the difficulty level of implementing this project.

2 Background

To detect lines and circles for border detection as well as ball detection, Hough transform is used after applying edge detection methods on the image. Hough transform is a method for isolating features of certain shapes from an image. According to a study in 2013 [1], applying Hough transform after using edge detection method significantly raised the accuracy of line detection.

According to another study for tracking basketball players[2], converting an image from BGR to HSV (hue, saturation and value) color model and performing erosion and dilation on the H-plane of the image before using Hough transform will increase the accuracy of detection. We implemented and examined 3 different approaches for player

detection. The first approach is detecting contours after filtering images using color threshold. The second approach is using histograms of oriented gradients. The third approach is using a single shot detector. In the end, we used the histograms of oriented gradients method in our final deliverable for player detection. As it has the best accuracy for human detection according to a study in 2005[3].

3 Approach

For detecting the borders of the court, first, we converted the image from BGR to HSV color model. We then extracted the H-plane and obtained the binary model of the image. After that, we performed edge detection using the Canny edge detector on the image and the Hough transform to mark all the lines. The end progress is showing in Figure 1. The green lines represent lines detected by Hough transform.

Figure 1Example of result after applying Hough transform for border detection

Though the approximate position of the upper and lower borders are marked after performing Hough transform, all the marked lines need to be analyzed to get the actual location of upper and lower borders. This was done by grouping all the lines by their slope and intercept point on Y axis. The number of lines in the group representing the actual upper and lower borders is the biggest among all other groups.

For the left and right borders, as the camera angle is changing, the intercept point on the Y axis for the left and right borders is also changing. But the slope of them remains the same. Thus, we first calculate the detected left borders' slope with an example result showing in Figure 2. Since there should be one left or right border present in the court, only the leftmost or rightmost line is picked up and marked as a border.

Figure 2 Getting the left border's slope

For the detection of the ball and players, we tried 3 different approaches. First is detecting players by changing the HSV boundaries and detecting color blocks. The example of the result is displayed in Figure 3. Using this approach, we first convert the image from BGR to HSV color model. Then we change the HSV threshold constantly until we can detect remaining color boxes that represent the players.

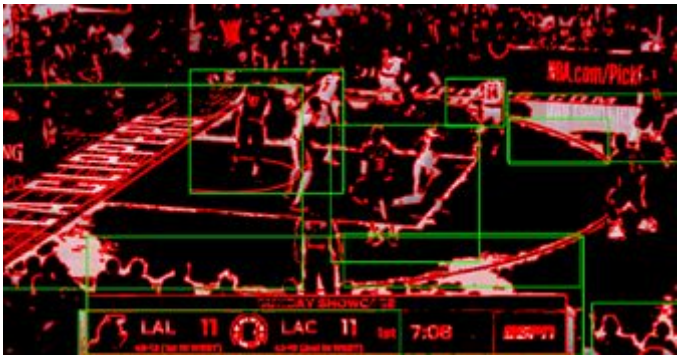

Figure 3 Detecting players by detecting color box

Then, we tried using a histogram of oriented gradients to detect the players following the instructions on studies [2][3]. Using this method, we were able to achieve a better result compared to detecting color boxes.

Finally, we tried using single shot detection for detecting players. As the project is able to recognize an object and label the frame of the object. It is important to have a large enough database which stores the trained models of a specific object[4]. And the object will appear in the video, such as balls and players. To analyze objects which appear in an image, it will extract some features which could represent a specific object. To apply extraction of features. The program will apply a certain convolution to the image which intends to obtain a feature layer of the object, and the feature layer will have p channels. In every location of an image, there will be a certain size of bounding boxes which will be used to achieve high accuracy of detections[5].

4 Result

After combining our work, the project is able to detect the borders, players and the basketball in real-time. We also implemented trackers shown in Figure 4 for the user to change some of the threshold used for detection and analyzation, as well as the special effects to be applied on the video.

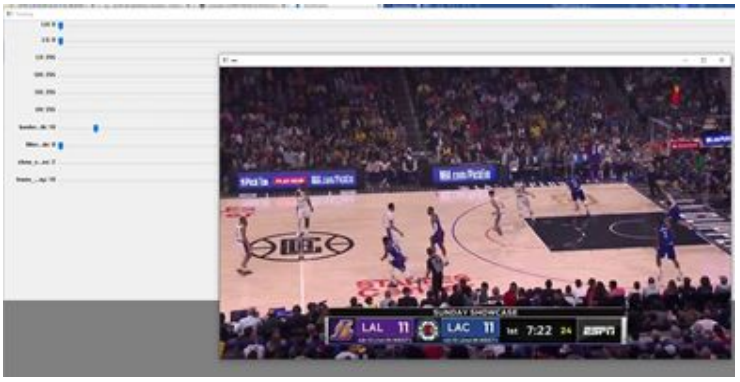

Figure 4 Displaying result video and tracker

Once our analyzing algorithm detects a moment shift, a fast break for example, special effects will be applied on detected players showing in Figure 5.

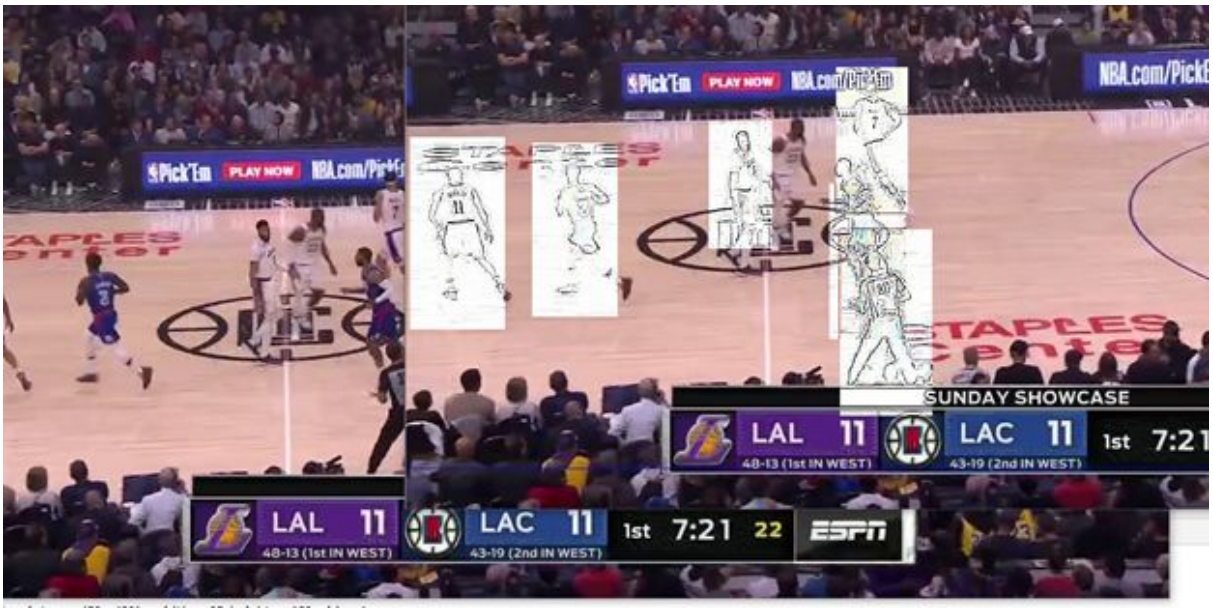

Figure 5 Special effects applied on detected players during a fast break

During the making of the project, we faced lots of difficulties. One of the biggest issues we had is the camera angle is changing all the time. As most of the detection algorithm relies on a stable camera angle, we had to develop an algorithm setting up conditions to filter the detected result. This, however, significantly increased the process time needed for the project. There is a noticeable delay of the video.

A future improvement for addressing this issue would be calculating camera angle shift first. Then calculating objects movement speed. Use the object speed together with the calculated camera angle shift, the next possible location of the objects can be pre-calculated. Thus the total process time can be reduced and the accuracy can be improved.

5 List of Work

Equal work was performed by both project members.

6 GitHub

https://github.com/ChenWangCarleton/COMP4102FinalProject

7 Reference

[1] S. Singh1 and A. Datar2, “EDGE Detection Techniques Using Hough Transform.” [Online]. Available: https://pdfs.semanticscholar.org/6dfd/9a799dd18a05b417302a30660fc7302cff52.pdf. [Accessed: 17-Apr-2020].

[2] E. Cheshire, C. Halasz, and J. K. Perin, “Player Tracking and Analysis of Basketball Plays.” [Online]. Available: https://web.stanford.edu/class/ee368/Project_Spring_1415/Reports/Cheshire_Halasz_Perin.pdf. [Accessed: 17-Apr-2020].

[3] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 2005, pp. 886-893 vol. 1.

[4] “Training Custom Object Detector¶,” Training Custom Object Detector - TensorFlow Object Detection API tutorial documentation. [Online]. Available: https://tensorflow-object-detection-api-tutorial.readthedocs.io/en/latest/training.html. [Accessed: 16-Apr-2020].

[5] S.-H. Tsang, “Review: SSD - Single Shot Detector (Object Detection),” Medium, 11-Nov-2019. [Online]. Available: https://towardsdatascience.com/review-ssd-single-shot-detector-object-detection-851a94607d11. [Accessed: 16-Apr-2020].