Original Article

ISSN: 2349 - 4891

International
Journal of Recent Research and Applied Studies
(Multidisciplinary Open Access Refereed e-Journal)

Basketball Player Tracking and Automated Analysis

R.Shakila[1], Dr. G.Nallavan[2]

[1]*M.Tech Student, Department of Advanced Sports Training and Technology, TamilNadu Physical Education and Sports University, Chennai, India.*
[2]*Assistant Professor, Department of Advanced Sports Training and Technology, TamilNadu Physical Education and Sports University, Chennai, India.*

Received 8th April 2015, Accepted 15th June 2015

Abstract

The goal for the image processing software is to perform Detection of each basketball player on the court, Team classification, and Individual player tracking. We first plan to detect each individual player on the court while minimizing the detection of non-players (crowd and referees). The video frames of the game will come from a static camera view that will not change. Next, we'll classify each player as either Team A or Team B based on a unique identifier, such as jersey color. Finally, we will try to maintain a track on each player so that their position can be stored and analyzed. Difficulties may arise as players move around each other on the court and cross paths. A stretch goal includes displaying colored dots overlaid onto a court image that shows each player's team and current position, as well a statistics based on each player's location. The player detection and tracking software will be implemented in MATLAB 2013a (Mathworks). we were able to accurately detect and track the individual players until more complex situations arose, such as players overlapping on the court. In ideal situations, these techniques provided reliable detection and tracking.

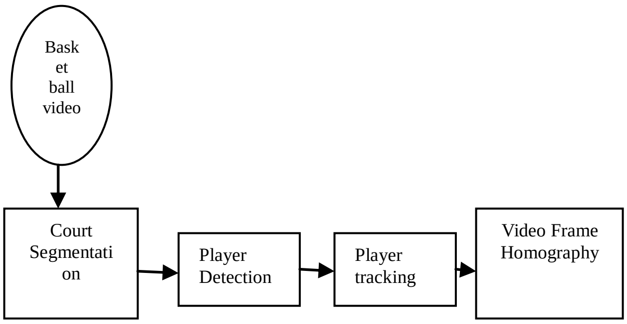
Keywords: Tracking, Image Processing, Basketball, Classification, Homography Technique.

Introduction

The problem of automatic tracking and identification of players in broadcast sport videos shot with a moving camera from a medium distance. While there are many good tracking systems, there are fewer methods that can identify the tracked layers[1]. Player identification is challenging in such videos due to blurry facial features (due to fast camera motion and low-resolution) and rarely visible jersey numbers (which, when visible, are deformed due to player movements). Automated basketball player detection and tracking has many benefits for both professional and collegiate athletics. Automated statistics could provide teams information about their opposition's plays, formations, and strategy. Real-time image analysis could also enhance the current state of rule verification by removing the referees' human errors from the game. In addition, player tracking could improve video broadcasting by automatically switching to the camera with the best viewing angle, or by focusing on superstars[2]. This would reduce the manual workload, which is primarily how sports broadcasting is done today. This paper provides a Matlab program and techniques to detect each individual player, classify which team he is on, maintain a track on each player, and projects their position onto a top-down view of the basketball court. The Matlab program allows the user to view the raw video and the detected positions side by side for comparison.

Correspondence
R.Shakila
E-mail: shaki901113@gmail.com, Ph: +9198845 80548

Methodology



Court Segmentation

First, the video frame is binarized so that the court pixels are the foreground and all other pixels are the background. This step eliminates non-interest areas, such as the crowd. In order to perform this segmentation, a MAP detector is trained by using training masks on the first 10 frames of the video. By isolating the known court pixels over multiple frames, the MAP detector is trained to recognize the average RGB color values of the court pixels[3]. After determining the average RGB values, the MAP detector binary thresholds for pixels within10% of the averages. This results in a noisy, but clear outline of the court. Morphological operators fill in the black holes and smooth the edges to produce a clean, binary mask (Fig 1. right).

A common player position occurs many times at the edge of the court, so that his body is outside of the court boundaries, such as the top-right Ohio St. player (Fig 1. left). To mitigate removing these player's bodies from the image, the binary court image is dilated to expand the overall region. Another major source of noise comes from the scoreboard at the bottom of each frame[4]. This causes major issues since the scores are color coded to match the players jersey colors, which could produce false detections. However, since the scoreboard is in a static position in each frame, it can be filtered out with a binary mask.

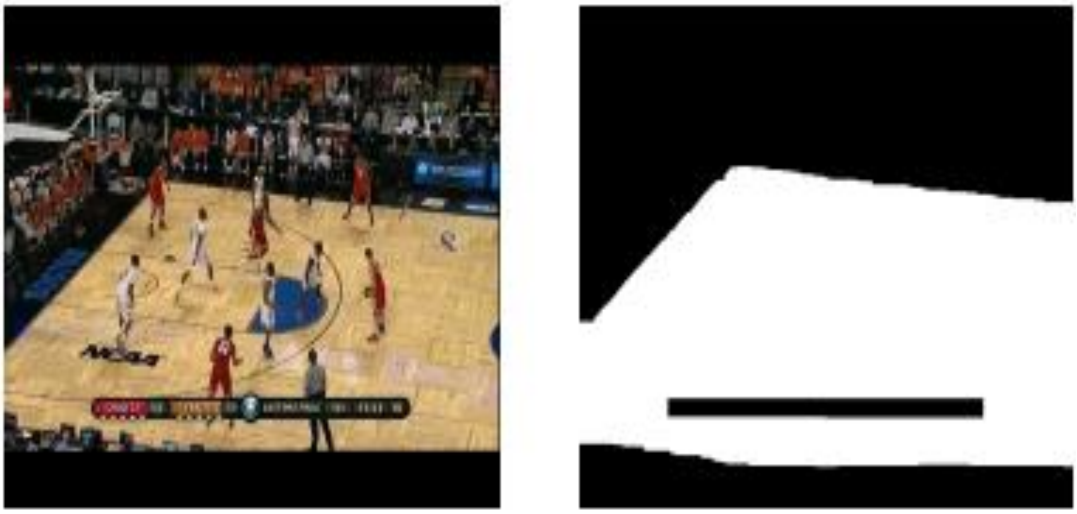


Figure I. The original first frame of the sample video (left) and the final binary court mask of the first frame (right)

Player Detection

The next step involves correctly detecting each individual player and classifying which team he is on. Similar to the court segmentation, a MAP detector is trained to recognize the average YCbCr values for each team's jerseys by using a set of training masks [3]. After training the MAP detector, image processing is applied to each frame for player detection. First, the image is multiplied by the binary court mask to remove the noise (Fig II. left). Then the image is binary thresholded based on the expected YCbCr values for each team (Fig II. right). Next, a morphological close operation using a 25x20 rectangular structuring elements increases the size of the largest detections (Fig III. left). A 25x20 structuring element is used because of its resemblance to the players, given that the players are taller than they are wide. Finally, the 10 largest, connected white blobs are detected as player positions for each team. The 10 largest instead of the 5 largest are chosen because the largest blobs do not always correspond to the correct players. By detecting more objects, the tracking function can filter based on track correlation (Fig III. right).

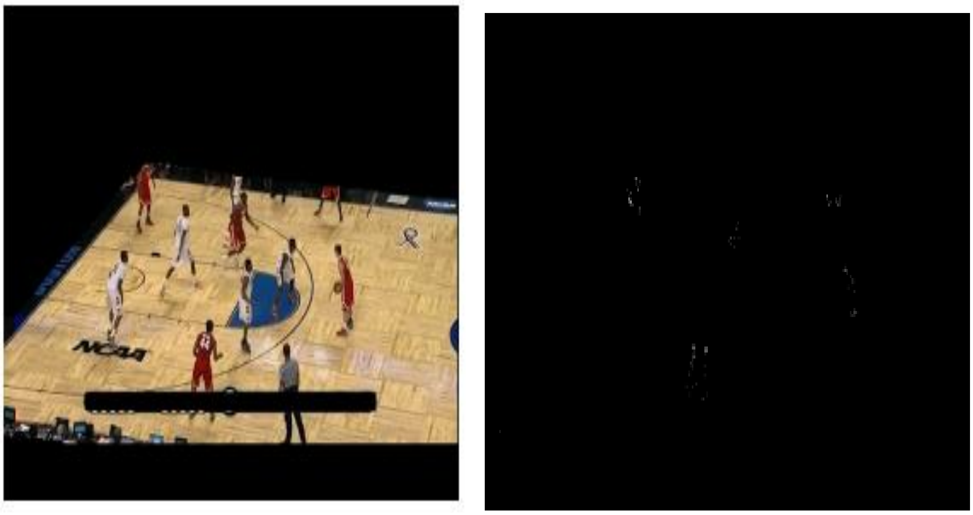


Figure II. A video frame multiplied by the binary court mask(left) and The binary court masked video frame after being threshold based on Ohio St.'s trained YCbCr values(right)

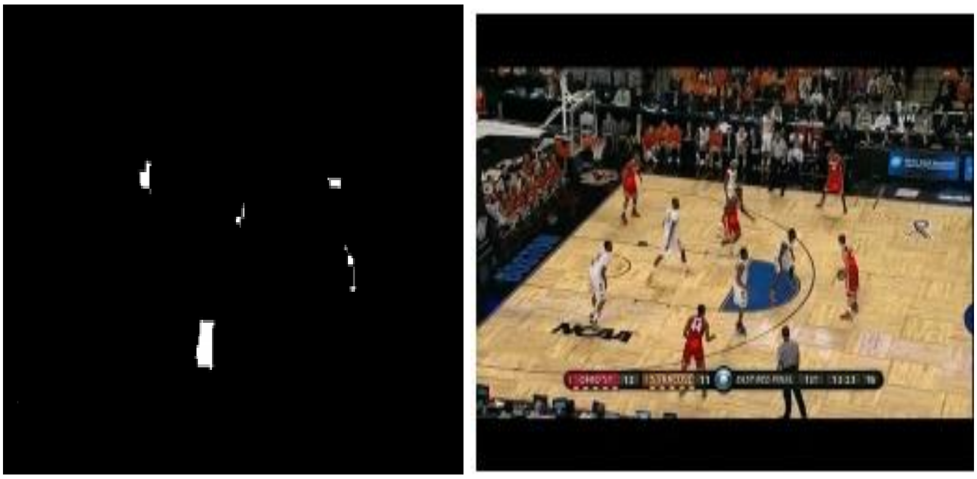


Figure III. The binary thresholded image after a morphological close operation: The original first frame of the sample video (left) and the final binary court mask of the first frame (right)

Player Tracking

In order to identify players, we have to first locate and track players over time, i.e., do multi-target tracking. This paper takes a tracking-by-detection approach, similar to. Specifically, we first run an object detector to locate players in every frame of a sports video, then we associate detections over frames with tracklets (a tracklet is a sequence of bounding boxes containing the same player over a period of time).

Once detected players have been separated into teams, the next step is to associate detected bounding boxes over time with tracklets. Here, we take a tracking-by-detection approach, where the inputs are detections and outputs are tracklets of players. We take a one-pass approach for tracking[5]. At any frame, we first associate detections with existing tracklets. To ensure a one-to-one matching between detections and tracklets, we perform bi-partite matching[6]. The matching scores are Euclidean distances between centers of bounding boxes and the predicted locations of players. We intentionally do not use colors in the matching score since players of a team wear the same uniform. After assigning detections to existing tracklets, the next step is to update the state estimate of players.

After possible players are detected for each team, the system establishes new tracks or correlates the current track to an existing one. For the first frame, the largest 5 connected areas are taken to be the correct positions. The player’s (x,y) pixel location is computed as the centroid of the connected area, plus 30 additional pixels in the y direction. This offset pushes the centroid down, which results in a pixel location that is closer to the player’s feet, rather than their waist. After the first 5 player detections in the first frame for each team, the system loops through the subsequent frames and performs track correlation. For each of the 20 detected possible players, it compares the (x,y) location to the 10 established tracks similar to player tracking in. If the Euclidean distance between the (x,y) locations is within 50 pixels and detections are for the same team, then the tracking function correlates the tracks and updates the current pixel location. If no player detection is found within 50 pixels of an already established track, then it’s previous (x,y) location is repeated in it’s track structure.

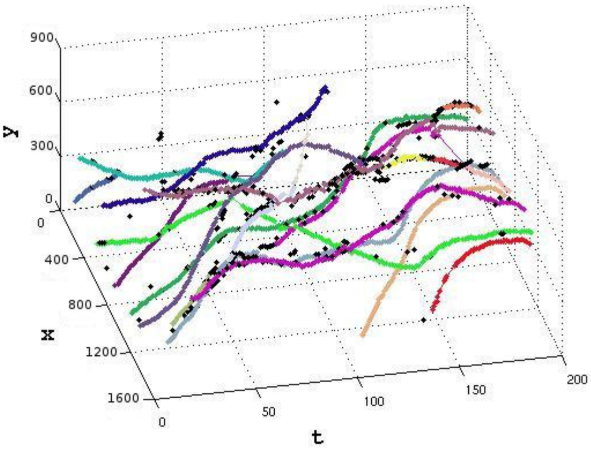


Figure IV. The x-y-t graph of tracking results, where (x; y) is the center of a bounding box and t is the time. Every dot in the graph represents a detected bounding box, where different colors represent different tracklets.

The state vector we want to track at time t is a 4-dimensional vector xt = [x; y;w; h]T , where (x; y) represents the center of the bounding box, and (w; h) are its width and height, respectively. Let zt = [x; y; w; h]T be the detected bounding box at time t. We assume the following linear-Gaussian observation model: p(ztjxt) = N(ztjxt;_z), where _z is a diagonal matrix set by hand. Most tracking systems assume a first-order or second order auto-regressive model for dynamics. That is, they assume that p(xtjxt□1) = N(xtjAxt□1;_x). More sophisticated models use Gaussian process regression to model the dynamics. We got much better results by using a simpler model of the form p(xtjxt□1; t) = N(xtjtat + bt;_x), where t is the current frame, at is a regression weight vector, and bt is a regression offset term[8]. The regression parameters (at; bt) are learned online based on a sliding window of data of the form (ti; ^xti), for ti = t □ F; : : : ; t □ 1, where ^xti is the posterior mean state estimate at time ti. Note that our motion model is independent of the previous state. However, it depends on the current time index. The reason it works well is that there is a local linear relationship between time t and the current state xt, as illustrated in Fig IV. Given our linear-Gaussian observation and motion models, we then update the current state using a Kalman Filter (KF).

We create a new tracklet for detections that are not associated with any existing tracklets. This new tracklet will be first marked as unreliable until it has a certain number of detections associated with it. Otherwise, the new tracklet will be automatically dropped. For tracklets that are not associated with any detection, the system will update the state using the prediction. However, if the tracklet does not have a detection over some time period (currently, 1 sec in experiments), it will be removed from the pool. The system will also terminate a tracklet when its bounding box moves out of the image border.

Video Frame Homography

The final step in the process is to project each player's frame position to their actual position on a top-down view of the court. A key assumption is that the video feed consists of a static camera angle, which does not require the homography matrix to be computed dynamically. By using this assumption, a single 3x3 homography matrix is pre-computed using an affine transform[9]. Each player's (x,y) pixel location is multiplied by the homography matrix, which projects their true position onto the top-down view of the court image.

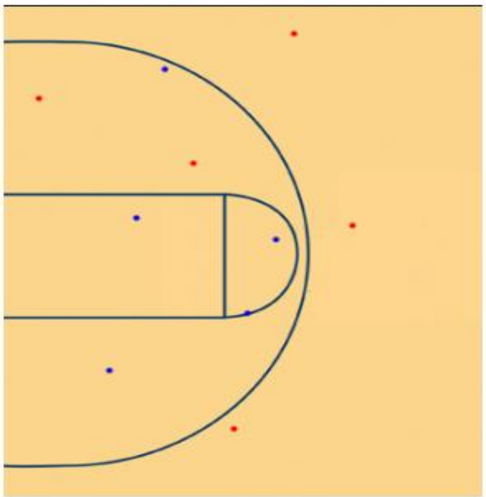


Figure V. The detected player positions for Ohio St. (red) and Syracuse (blue) after projection using an affine transformation

Results and Discussion

Our system was applied to various clips from 2 games, Oregon vs. Washington and Ohio vs. Syracuse during the 2011 NCAA tournament, recorded in 720p HD quality. The Fig. VI shows the court region and the paint lines are correctly detected.

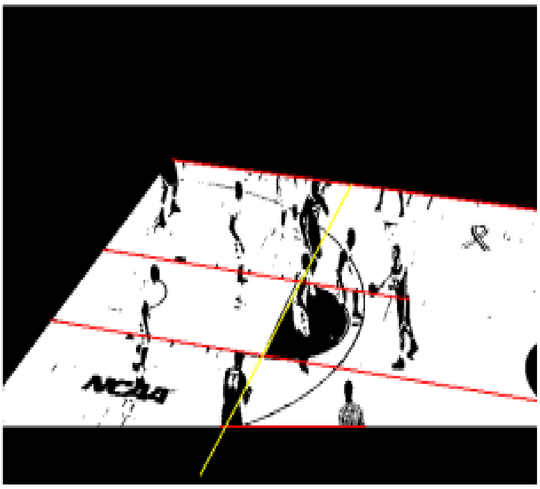


Figure VI. Court lines correctly detected using hough transform

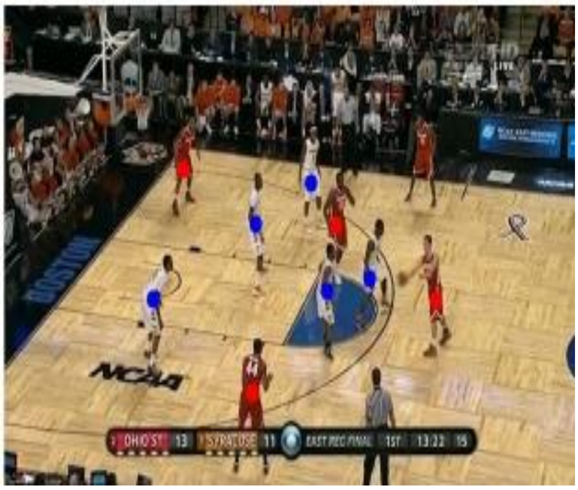


Figure VII. Player centroids detected and classified by team

In Fig. 7, we see that players are able to be detected and split into their respective teams. Also, we notice that players at the very top of the court region are not detected. From the Ohio video, all players within the court region are correctly detected in 67 out of 105 frames. In the Oregon video, players within the court region are correctly detected in 44 out of 86 frames. The reduction in detection rate is due to the higher amount of occlusion in the clip. Using our tracking algorithm on the Ohio State vs. Syracuse game, we see that player trajectories are accurate for all players detected in the first frame except for one. One of the Syracuse players overlaps with another and the two merge into one trajectory. Additionally the homography matrix that is responsible for translating the players 3D coordinates to 2D coordinates is slightly less accurate the further a player is from the key and assumes that players' centroids are directly on the floor. For this reason, the trajectories of players are frequently distorted as the number of frames in a given video increases. In the Oregon vs. Washington game, we met with the best results using frames 1 to 85 and frames 16 to 120 for the Ohio vs. Syracuse game

Conclusion

Future work can be processed with tracking the players by recognizing the letters printed on jersy, team logo in arm at the side angle and also by recognizing the individual player numbers. It may provide more insight and confidence studies for image processing in order to obtain results in all aspects; hence it improves accuracy and efficiency in results.

References

1. Anton Andriyenko, Konrad Schindler, Stefan Roth, "Discrete-Continuous Optimization for Multi-Target Tracking," Computer Vision and Pattern Recognition, IEEE (2012)
2. Ekin, A, Tekalp, A.M. , "Robust dominant color region detection and color-based applications for sports video," Image Processing, International Conference (2003)
3. J. Xing, H. Ai, L. Liu, and S. Lao, "Multiple player tracking in sports video: A dual-mode two-way bayesian inference approach with progressive observation modeling," Image Processing, IEEE Transactions (2011)
4. Jingchen Liu, Peter Carr, Robert T, Collins, Yanxi Liu, "Tracking Sports Players with Context-Conditioned Motion Models," Image Processing, IEEE (2013)
5. Min-Chun Hu; Ming-Hsiu Chang, "Robust Camera Calibration and Player Tracking in Broadcast Basketball Video," IEEE Transactions (2011)
6. Seung-Hwan Bae, Kuk-Jin Yoon, "Robust Online Multi-Object Tracking based on Tracklet Confidence and Online Discriminative Appearance Learning," Global Frontier R&D (2013).
7. Tianzhu ZhanG, Narendra Ahuja, "Robust Multi-Object Tracking via Cross-Domain Contextual Information for Sports Video Analysis," , Image Processing, IEEE (2012)
8. Tomita, A. Echigo, T, Knrokawa, M., Miyamori, H, Iisaku, S, "A visual tracking system for sports video annotation in unconstrained environments," Image Processing, (2000)
9. Tsung-Sheng Fu, Hua-Tsung Chen, Chien-Li Chou, Wen-Jiin Tsai, SuhYin Lee, "Screen-strategy analysis in broadcast basketball video using player tracking," Visual Communications and Image Processing, IEEE (2011).

Please cite this article as: R.Shakila, Dr. G.Nallavan. Basketball Player Tracking and Automated Analysis. *International Journal of Recent Research and Applied Studies*, 2015, 2, 6(10), 42 -46.