Top view stitching and tracking (tracking and geometry)

Computer Vision Project

Pietro Bologna, Christian Sassi

Department of Information Engineering and Computer Science at the University of Trento

Abstract

This project focuses on processing video footage captured at the Sanbapolis facility in Trento. It utilizes camera stitching techniques to combine multiple views into a unified top-down representation of the volleyball court. Object detection and tracking are applied to both the players and the ball, while color-based team identification is used to distinguish between teams during player tracking. All techniques are implemented in real-time, allowing for immediate visualization of the results.

I. Introduction

Before starting, all raw videos were pre-processed by synchronizing them manually and removing irrelevant scenes via a dedicated script. Once prepared, the files proceeded through the following tasks:

1. **Top-View Court Stitching**: The facility comprises three views—*top, center, and bottom*—each captured by four cameras. Images were stitched within each view and then combined to form a seamless top-view representation.
2. **Object Detection**: Various detection methods, including frame subtraction, background subtraction, adaptive background subtraction, and Gaussian averaging, were tested to the stitched images.
3. **Object Tracking**: A particle filtering technique was used to track detected objects.
4. **Ball Detection and Tracking**: YOLO was employed for ball detection due to its accuracy, while particle filtering was adapted for tracking.
5. **Color-Based Team Identification**: The net was used as a natural boundary to differentiate teams, as uniform colors made traditional identification impractical.

II. Top-View Court Stitching

The stitching process began by combining images from the same view, each captured by four cameras. The left-most and right-most views were discarded as they captured irrelevant areas. The remaining two views were stitched using their shared regions, manually defined due to fixed camera positions, to ensure accurate feature matching.

SIFT was applied to detect keypoints, and a brute-force matcher paired them. A ad-hoc filtering algorithm removed incorrect matches by analyzing feature pair inclinations before using OpenCV's `findHomography` function. While individual view stitching produced accurate results, combining all views also required manual selection of key pairs for robustness. Fig. 1 illustrate the final stitched image.

Figure 1: *Stitched image.*

However, a detailed inspection reveals artifacts caused by camera angles, where feet align correctly but upper body misalignment occurs, as showed in Fig. 2. The green circle shows that the feet align correctly, while the red circle highlights a misalignment in the upper body. This artifact is a result of the view angle effects of the cameras.

Figure 2: *Camera view angles causing artifacts (left) and example of stitching artifacts affecting the player (right).*

Finally, to enhance real-time performance, a caching system was implemented to store stitching parameters, eliminating the need for recalculations.

III. Object Detection on Top-View Images

Starting from the result obtained through stitching, various detection algorithms were tested, including frame subtraction, background subtraction, adaptive background subtraction, and Gaussian averaging. Each method was implemented and tested, leading to the following observations:

- **Frame subtraction**: Ineffective for stationary objects; increasing the temporal window worsens detection quality.
- **Background subtraction**: Reliable due to a stable background, detects both moving and stationary objects.
- **Adaptive background subtraction**: Similar to background subtraction but adapts to lighting changes. Large alpha values fail for stationary objects, so smaller values were used.
- **Gaussian averaging**: Effective for stable backgrounds with minor changes. A small alpha value made it nearly identical to background subtraction.

Background subtraction was found to be the most effective due to the stable background, allowing reliable detection even for stationary objects. For this technique, a manually selected background frame ensured consistency throughout the video. Post-processing involved applying a threshold and dilation to correct stitching artifacts, merging fragmented player detections and obtaining the result shown in Fig. 3.

Figure 3: *Thresholded and dilated image.*

Then, objects were filtered using court boundaries, removing irrelevant detections (e.g., coaches stepping into the frame). Finally, by combining these techniques, the result shown in Fig. 4 was obtained.
Despite these optimizations, occasional inaccuracies occurred, such as merging multiple players into a single detection. A trained model could further improve differentiation, but the idea was to use as much course-related techniques as possible.

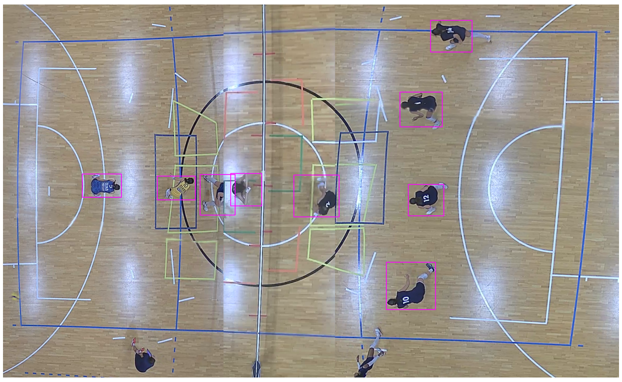

Figure 4: *Motion detection.*

IV. Object Tracking

A particle filtering approach was implemented iteratively, testing one method at a time until a suitable technique was found. Particle filtering performed well, making further exploration unnecessary. Each detected object was assigned to a particle system, which initially exhibited chaotic behavior due to random initialization, as shown in Fig. 5.

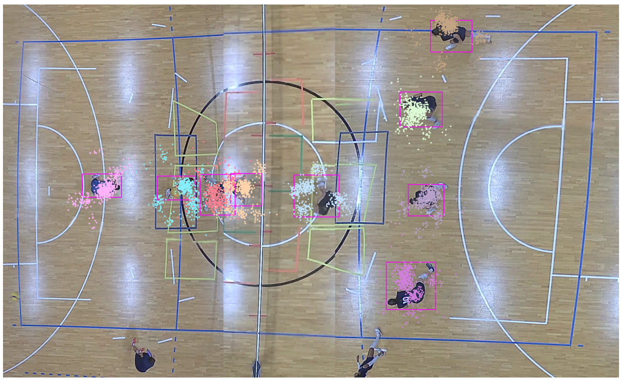

Figure 5: *Initial particle systems.*

At each step, particle systems were matched with updated bounding boxes based on centroid distance. If a match was found, the system was linked to the corresponding object; otherwise, a new particle system was created. This allowed continuous tracking of moving objects. Additionally, as the system iterated, randomness decreased, leading to more stable tracking. In the end, the results shown in Figure 6 were achieved. The system predicted movement direction, but sudden and rapid changes caused temporary inaccuracies. Additional challenges arose when objects were lost and later re-detected, as this created new particle systems, leading to unstable tracking, as happened at the beginning. Similarly, when multiple players were merged into a single detection, the system either reassigned an existing particle system or generated a new one, causing brief tracking inconsistencies. Despite these limitations, particle filtering

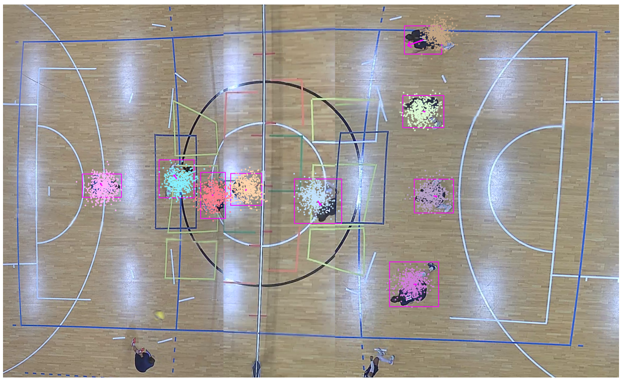

Figure 6: *Motion tracking.*

effectively tracked player movements. While other techniques could enhance stability, this method was sufficient for the project's objectives.

V. Ball Detection and Tracking

Initially, previous detection techniques were tested, but the ball's high speed and distortion made detection challenging. Instead, YOLO v11 was adopted, using a custom dataset of ~1000 manually labeled images created with Roboflow [1]. A confidence threshold above 50% ensured accurate detection.

For tracking, a modified particle filtering approach was used. Unlike player tracking, the same particle system persisted throughout the video, eliminating the need to reassign bounding boxes. If the ball temporarily disappeared, the system cached its position and resumed tracking upon reappearance. This approach produced the results shown in Fig. 7. While

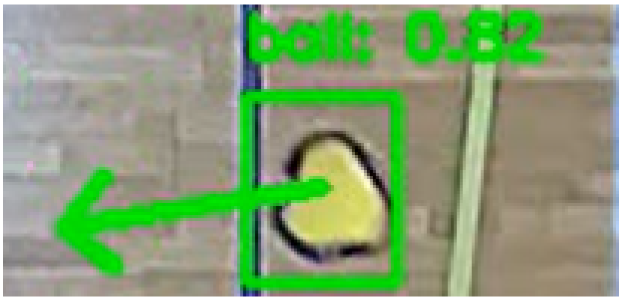

Figure 7: *Ball detection and tracking.*

sudden movements required brief adjustments, the ball's predictable trajectory made this method highly effective.

VI. Color-Based Team Identification

For this task, the net was used to separate teams, as volleyball players do not mix like in other sports. In fact, color-based identification was challenging since both teams had similar uniforms, as shown in the histograms in Fig. 8. The first two histograms highlight the similarity between team uniforms; however, the third histogram shows that differentiation is possible when uniforms have distinct colors (e.g. *libero* player).

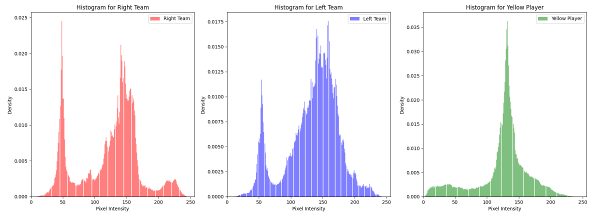

Figure 8: *Color histograms of player uniforms.*

The method assigned players to teams based on their position relative to a fixed X-coordinate, making it computationally efficient. However, when multiple players near the net merged into a single blob, misclassification could occur. YOLO could improve detection accuracy, but the project prioritized techniques covered in the course.

VII. Results

The project successfully developed a volleyball video analysis system using multi-camera setups. While generally effective, some limitations remain.

- **Video Stitching**: The final stitching results, though accurate, contain visual artifacts due to camera angles. A wide-angle camera could reduce these issues.
- **Object Detection**: Background subtraction was the best-performing method among the tested techniques. However, its inherent limitations might be overcome using more advanced approaches like YOLO.
- **Object Tracking**: Particle systems were used to maintain object associations, despite implementation challenges. The results were pretty satisfactory.
- **Ball Detection & Tracking**: Traditional methods struggled due to the ball's speed and size, so YOLO was employed for better accuracy.
- **Team Identification**: Players were separated using the net, but detection inaccuracies sometimes merged players from both teams into one object. YOLO could improve this aspect.

The project builds on modern computer vision techniques for multi-camera stitching, object detection, and tracking, aligning with recent advancements in sports video analysis.

[1] Roboflow: https://roboflow.com/