Cross-Camera Player Mapping: A Technical Report

Candidate: Chiranjan Yathish (github)
Date: 27th June, 2025

---

1. Objective

The goal of this project was to develop a system capable of detecting and mapping football players across two synchronized video feeds from different camera perspectives: a wide-angle, top-down **tacticam** view and a standard, angled **broadcast** view. The primary objective was to assign a single, consistent, and persistent player_id to each individual, ensuring their identity is maintained across both video streams.

---

2. My Approach and Methodology

I designed a modular, multi-stage pipeline to tackle this complex computer vision problem. The core challenge lies in bridging the geometric and appearance gap between the two views. My methodology addresses this by combining geometric transformation with robust tracking and feature-based matching.

The final pipeline consists of four main stages:

Stage 1: Detection and Filtering
The foundation of the system is accurate player detection. I utilized the provided YOLOv8 model to identify player bounding boxes in each frame. Recognizing that raw detector output can be noisy, I implemented a crucial **Non-Maximum Suppression (NMS)** filter. This post-processing step intelligently removes redundant, overlapping detections for the same player, ensuring that the subsequent tracking and mapping stages receive clean, unambiguous data.

Stage 2: Geometric Alignment via Homography
To establish a common ground for player positions, I used a **homography transformation**. A 3x3 homography matrix was calculated to map pixel coordinates from the broadcast view's ground plane to the tacticam view's ground plane. This was achieved by:

1. Creating a utility script to manually select corresponding, static reference points on the pitch (e.g., corners of the 6-yard box) in both video feeds.
2. Using OpenCV's cv2.findHomography to compute the transformation matrix.
3. The player's anchor point for this transformation was deliberately chosen as the **bottom-center of their bounding box**, as this most accurately represents their contact point with the pitch.

Stage 3: Intra-Video Tracking
To handle temporary occlusions or detection failures, I built a Tracker class to maintain stable player IDs *within* each video stream. This tracker uses the **Hungarian algorithm** to optimally match players between consecutive frames based on positional proximity, creating "sticky" temporary IDs (e.g., tacticam_player_1, broadcast_player_8).

Stage 4: Advanced Cross-Camera Mapping and Re-identification
This is the core intelligence of the system, encapsulated in a dedicated PlayerMapper class. This class performs the final, global ID assignment by:

1. **Transforming** the positions of broadcast players into the tacticam's coordinate space using the homography matrix.
2. **Creating a hybrid cost matrix.** Instead of relying solely on position, the cost to match two players is a weighted combination of:
    - **Spatial Distance:** The Euclidean distance between the players in the common tacticam coordinate space.
    - **Appearance Similarity:** The similarity of their **HSV color histograms**, which captures their jersey color and provides a powerful feature to distinguish between nearby players.
3. **Optimal Assignment:** The Hungarian algorithm is used again to find the best pairings that minimize this hybrid cost.
4. **Temporal Persistence:** The mapper maintains a memory of recently lost players. If a player reappears, the system attempts to re-identify them based on their last known position and color profile, ensuring long-term ID consistency.

---

3. Techniques I Tried and Their Outcomes

Several techniques were evaluated and iterated upon to arrive at the final, robust solution.

- **Initial Technique: Purely Positional Matching**
    - **Description:** My first attempt involved using only the homography-transformed positions to match players.
    - **Outcome:** This worked reasonably well for isolated players but failed significantly when players were clustered together. The system would frequently swap IDs between teammates standing close to each other, as their spatial distance was nearly identical. This demonstrated that positional information alone is insufficient for high accuracy.

- **Refined Technique: Hybrid Spatial + Color Matching**
    - **Description:** To solve the ambiguity of close players, I introduced appearance features by calculating HSV color histograms for each player's bounding box. The matching cost was then updated to be a weighted sum of spatial distance and color dissimilarity.
    - **Outcome:** This was a major success. The system could now easily distinguish between players on different teams (e.g., red vs. white jerseys) even when they were side-by-side. This dramatically reduced ID-swapping and significantly improved the overall mapping accuracy.

- **Initial Technique: Stateless ID Assignment**
    - **Description:** The initial mapping logic was stateless, meaning it only considered the information available in the current frame.
    - **Outcome:** This led to "flickering" IDs. If a player was occluded for even a few frames, they would be assigned a completely new ID upon reappearance.
    - **Solution:** This was addressed by engineering the PlayerMapper to be stateful. By adding a memory component that tracks lost players and their features, the system gained the ability to perform **re-identification**, leading to far more stable and persistent player IDs over time.

---

4. Challenges Encountered

- **Noisy Detections:** The object detector occasionally produced multiple, overlapping bounding boxes for a single player. This noise confused the tracker, leading to multiple temporary IDs being assigned to one individual. This was effectively solved by implementing the NMS filter before the tracking stage.
- **Video Synchronization:** I discovered that the two video files were not perfectly synchronized from the first frame. I had to create a small utility to find a "golden frame" where the action aligned, and then applied a frame offset to the videos to ensure the mapping was performed on correctly synchronized moments of gameplay.
- **Static Homography Limitations:** The largest inherent challenge is that the calculated homography is static. It is only perfectly accurate as long as the broadcast camera does not significantly pan, tilt, or zoom. While sufficient for the provided clips, this is a limitation for a full-game implementation.

---

5. Future Work and Potential Improvements

If given more time and resources, I would focus on elevating this proof-of-concept to a production-grade system. My priorities would be:

1. **Dynamic Homography:** I would replace the static matrix with a dynamic one. This could be achieved by using feature-matching algorithms (like ORB) to automatically detect stable points (pitch lines) in each frame and recalculate the homography whenever a significant camera movement is detected.
2. **State Estimation with Kalman Filters:** I would upgrade the tracker to use Kalman Filters. This would provide smoother player trajectories by incorporating velocity and acceleration into the state estimation, leading to better predictions during brief occlusions and more robust tracking overall.
3. **Learning-Based Re-ID:** To further improve matching, especially between players on the same team, I would replace the color histogram with a deep learning-based Re-ID (Re-identification) model. This would extract a powerful feature embedding for each player, providing a more robust signature that is less sensitive to lighting and pose variations.