Homography based Player Identification in Live Sports

Yash Pandya Kaustav Nandy Shivam Agarwal
Amazon Prime Video
(yaspan, kaustn, agarshi)@amazon.com

Abstract

Modern live sports broadcasts display a wide variety of graphic visualizations identifying key players in a particular play. Traditionally, these graphics are created with extensive manual annotation for post-match analysis and take a significant amount of time to be produced. To create such visualizations in near real-time, automatic on-screen player identification and localization is essential. However, it is a challenging vision problem, especially for sports such as American football where the players wear elaborate protective equipment. In this work, we propose a novel approach which uses sensor data streams captured by wearables to automatically identify and locate on-screen players with low latency and high accuracy. The approach estimates a field registration homography using on-field player positions from RFID sensors, which is then used to identify and locate individual players on-screen. Experiments using American football data show that the method outperforms a deep learning based state-of-the-art(SOTA) vision-only field registration model both in terms of accuracy of the homography and also success rate of correct homography computation. On a dataset of over 150 replay clips, the proposed method correctly estimated the homography for approximately 25% additional clips as compared to the SOTA method. We demonstrate the efficacy of our method by applying it to the problem of rendering visualizations around key players within a few minutes of the live play. The player identification accuracy for these key players was over 96% across all clips, with an end-to-end latency of less than 1 minute.

1. Introduction

In recent times, the use of captivating visualizations to enhance the sports broadcast viewing experience has become extremely popular. These features help improve engagement of the viewers by enabling better comprehension of the game. As compared to the in-stadium viewing experience, in quite a few cases, it is difficult to understand the game strategies and proceedings while viewing the same on a smaller screen. The visualizations (Fig. 1) make it easier to understand the game by highlighting key players, their movements, actions, on-field landmarks and key moments in the game. One of the first and most successful such system was the "1^{st} and Ten" [1] visualization which, with the help of 3D modelling algorithms and camera hardware, augmented the viewing experience of American football by adding a yellow line at the first down. During a live game, low latency and high accuracy are essential for the visualizations to capture viewer attention to the fullest and computer vision (CV) algorithms play an important role in achieving that.

Figure 1. Sample visualization in American football.

With the advent of embedded sensors in player wearables and equipment, various auxiliary information streams are available directly from the playing field. They provide play metadata such as the nature of play, on-field player and/or ball locations, etc. These streams in conjunction with the CV algorithms can play a significant role in creating viewer-engaging graphic visualizations with very low end-to-end delivery time. For example, in the case of National Football League (NFL), RFID sensors embedded in the ball and player shoulder pads provide a continuous data stream of 2D coordinates of the entities on the playing field. Such information can augment the CV algorithms to display information such as route-map or animated graphics tracking key player locations on-screen and detection of play formations. However, there are several challenges in using the auxiliary information streams which need to be resolved before they can be faithfully used. One of them is to have a reliable and accurate synchronization of the visual/audio and auxiliary data streams so that temporal events in the video can be accurately aligned with the time stamps. Also, in order to use the player location data, mapping of on-field to on-screen locations is crucial.

In this paper, we describe our work on fully automatic on-screen player localization and identification utilizing the auxiliary data stream providing on-field player locations collected via RFID tags worn by players. It first performs field registration by estimating a perspective transformation from the ground plane to the display frame and then utilises helmet tracks that are created by a custom trained model, fine-tuned on an American football helmet dataset, to perform player localisation and identification. Unlike the conventional methods for field registration, this method does not depend on the identification of salient on-ground landmarks/key-points and mapping them to corresponding points on a ground template to estimate the image to template transformation. Although the techniques and overall workflow have been developed and tested for American football, the proposed solution can be extended to other sports given the availability of on-field player positions data. The major contributions of the paper are:

- To the best of our knowledge, this is the first work to report the usage of RFID on-field positional data for field registration and on-screen player identification.
- A method for automatic coarse synchronization of the visual and auxiliary data streams.
- A metric to evaluate the quality of field registration without ground truth data by using on-field player locations.
- Efficient extension of field registration homography to work for video clips in near real-time using traditional CV methods or on-field player location information.

2. Related Work

Automatic identification of players [25, 31] is a challenging problem in large field multiplayer sports. The most common methods perform player identification by detecting and recognizing the team and jersey number [2, 14, 16] of individual players from the appearance of their jerseys. In the case of sports where the players are not wearing protective gear, such as soccer and basketball, facial and appearance features can be used to identify the players [17, 18, 23]. For American football and ice hockey, the protective gear worn by the players makes it visually challenging to distinguish between players from the same team.

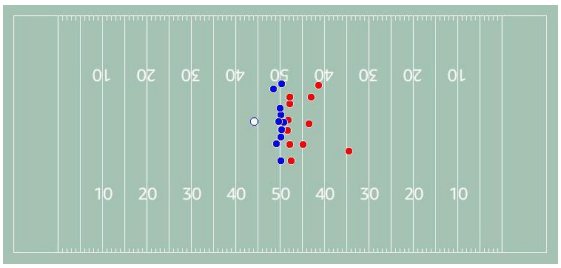
Sports field registration has been an active area of research both in academia and industry. Most CV based methods [7, 10, 12, 20, 27, 28] estimate the homography by identifying sports specific on-field landmarks which are used as key-points to estimate the homography using techniques such as RANSAC [8]. Early approaches [7, 12, 27, 28] which rely on classical computer vision use techniques such as Hough transforms [11], SIFT [15], and ORB [22], which were popular for the problem of camera calibration. Recent approaches [10, 20] use different deep learning techniques to either identify these keypoint features [20] or to directly regress homography matrix parameters [6]. Most of these approaches rely in some way on the field markers and might struggle in scenarios where these are sparse or missing. This often happens when the camera zooms in on the players in a small part of the field which is away from the edges. Some approaches [9, 21] try to propagate homography between consecutive frames when manual initialisation is available for a few frames. These methods can be used to overcome challenges related to less visibility of markings but even they need the stretch of frames with low confidence results to be short. [24] proposes a self-supervised data mining based method for registering cross modality images such as natural image and its edge map using a score regression network. [4] also proposes a key point based method but formulates that as an instance segmentation problem. There are various approaches adopted in the industry which rely on human workers or special expensive camera/hardware equipment to estimate the camera pose. Companies such as Stathletes generate homography for every frame of the game with the help of manual annotations by human workers.

In terms of the usage of sensor data from wearables, Carey et. al [3] report a method for improving player tracking results in Australian football. Although [5] utilize player location information to improve their camera parameter estimation algorithm, to the best of our knowledge, this is the first work which uses on-field player positions extracted through RFID sensors to estimate player identities by computing a homography for field registration. In contrast to other methods which rely on field markings, the proposed method is robust in handling poor field landmark visibility, but needs at least four players to be visible on the video frame to be able to estimate the homography. For our particular use case, this is guaranteed because we want to estimate the homography at the start of the play in American football where the camera includes most of the on-field players. Our paper focuses on computing the homography for one key frame which is subsequently propagated across frames with low latency using player/key-point tracking techniques.

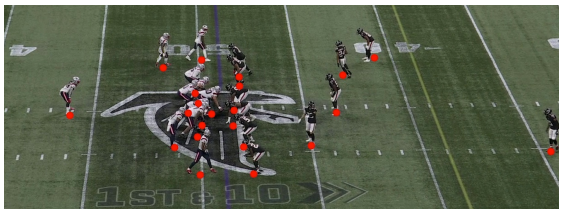
3. Methods

In this section, we describe proposed approaches to automatically identify players on-screen. We leverage a unique feature provided by NFL called Next Gen Stats(NGS). It contains key players involved in a play along with position, speed, and acceleration information for all players and the ball. The tracking information is generated by RFID chips embedded in the shoulder pads of the players and provides a view (Fig. 2a) of the on-field player positions.

A challenge in using this information is the absence of a common timestamp between the video and the NGS data to synchronise them. Both video frames and the NGS data



(a) NFL NGS data visualized as a 2D map for the frame of snap. The home and away team players are represented by red points and blue points respectively. The player in white is a player of interest who is the last ball carrier for the play.



(b) All player locations from the NFL NGS data mapped to the frame of snap and represented by red points in the replay clip using the homography computed by our solution.

Figure 2. Visualization of the NFL Next Gen Stats(NGS) tracking data for the frame of snap for a replay clip.

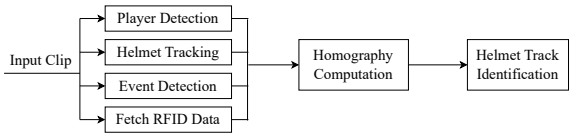


Figure 3. Steps in our proposed solution for automated detection of players in a replay clip from NFL.

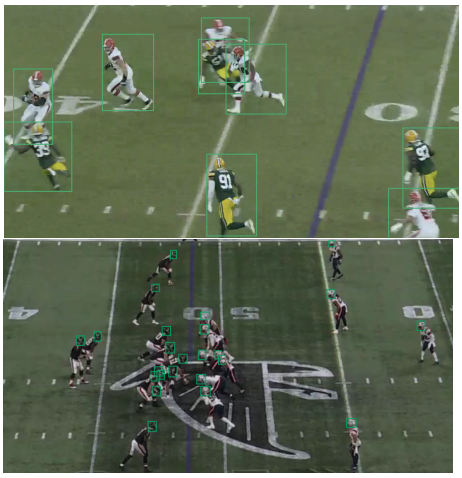


Figure 4. Visualization of the outputs of the player detection(top) and helmet detection(bottom) modules.

arrive with monotonic timestamps with fixed frame rate of 30 FPS and 10 FPS respectively. However, they use different timestamps and cannot be synced in a straightforward way since the replay clips are based on the broadcast stream which is not in sync with the on ground clock. To solve this issue, we identify common events in both the replay clip and the NGS data which contains timestamps for events such as snap (start of play), pass completion, touchdown, etc.

Task	mAP.5	mAP@.5:.95	Latency (FP16)
Player Detection	0.963	0.622	9.8ms
Helmet Detection	0.959	0.663	10.4ms

Table 1. Accuracy and latency metrics for the Scaled-YOLOv4 P5 model fine-tuned on our custom dataset.

Figure 3 shows a high-level view of our proposed solution targeted to solve the problem in replay clips. On receiving an input clip, four modules are triggered to synchronize the data streams and extract visual information. An **event detection** module predicts the frame in which the snap event occurs in the replay clip video. The **player detection** module provides player detection bounding boxes (Fig.4 - top) for each frame and the **helmet detection and tracking** module tracks player helmets as bounding box tracks (Fig.4 - bottom). The NGS data is used to identify players in the snap frame predicted by the **event detection** module. Player matching across NGS positional data and those detected in the video frame is performed using a homography or perspective transformation (computed by the **homography computation** module) mapping NGS on-field coordinates to points on the video frame. Such a player mapping is shown in Figure 2b. The **helmet track identification** module finds an on-screen player helmet track for a mapped point on the ground plane. The matched helmets can be tracked across frames using helmet tracking. This workflow can be used to power overlay visualisations (Fig. 1) used to highlight a player in live sports broadcast.

In some scenarios, such as helmet tracking failures and player exiting or re-entering the frame the homography needs to be recomputed for subsequent player identification. The proposed homography computation module can be re-used but at the cost of significant added latency. To handle such cases, we propose efficient low latency methods to extend the homography across a series of frames.

3.1. Player Detection and Helmet Tracking

Detection and tracking of players and helmets in American football broadcast is challenging due to excessive occlusions and motion blur (caused by sudden directional changes of both players and camera). A low latency yet accurate fine-tuned Scaled-YOLOv4 [26] model is used for both player and helmet detection. The key metrics for our fine-tuned detection models are captured in Table 1. Pre-trained data association based DeepSORT [30] [29] algorithm is used for helmet tracking. It uses detected bounding boxes as input and tracks the helmets using a Kalman filter based motion model and deep appearance features.

3.2. Snap Detection

An ensemble of deep action recognition models(TSM [13]) with different temporal support(7, 15, and 25 frames) is used for detecting the snap event. The models were trained to predict the probability of the snap event at each frame and were ensembled using the AND(multiplication) operation. The frame inferred to have the highest snap probability in a clip was predicted as the snap frame. Over a dataset of 160 replay clips the fine-tuned model predictions had a median deviation of 2 frames and P95 deviation of 5 frames from the ground truth snap frames. The proposed player identification approach could correctly identify a player of interest in the snap frame for over 96% of all clips which showed that it was robust enough to handle snap frame deviations of up to 6 frames.

3.3. Smart Sampling based Homography Computation

The player bounding boxes and NGS information for the predicted snap frame were used to compute a perspective transform from the NGS 2D plane to the video frame pixels. A perspective transform represented by a 3x3 matrix has 8 degrees of freedom and relates the points between two planes (up to a scale factor) in the homogeneous coordinate system:

$$s \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = H \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (1)$$

The 8 degrees of freedom is enforced by normalizing with $h_{33} = 1$ or $h_{11}^2 + h_{12}^2 + h_{13}^2 + h_{21}^2 + h_{22}^2 + h_{23}^2 + h_{31}^2 + h_{32}^2 + h_{33}^2 = 1$. Four or more corresponding points in the NGS plane and the video frame are required to compute the homography uniquely. The detected players and their position on the ground plane are utilized as the landmarks for computation. There are 22 player markers in the NGS 2D plane and a variable number of detected players in the video frame as shown in Fig. 5a.

As player correspondences between the NGS plane and the video frame are unknown, homographies are computed for a subset of all possible point correspondences. The homography which maps the points with minimal error is chosen as the final transformation. Iterating through all possible combinations of 4 matched points is exponential in complexity ($\approx {}^{22}C_4 {}^{22}P_4 = 1284221400$). Hence, we propose an intelligent sampling process to reduce the search space to a smaller set while still ensuring robustness.

Points in the NGS 2D plane with the minimum and maximum x and y coordinate values are removed to reduce 22 points to 18 points as some of these players might not be in the frame of the video while the broadcast camera zooms in to focus on the players closer to the ball. A set of points and their correct corresponding boxes is likely to generate a stable homography when they are well distributed across the ground plane of the video frame. These points should not be spatially co-linear or clustered in a small area. To ensure non co-linearity and spatial separation, 10 points are selected from the remaining 18 points by selecting 5 points with the highest and the lowest x coordinate values which removes the players very close to the ball who are also very close to each other. 8 player boxes are similarly selected from the snap video frame. The selection of the NGS points and player boxes is visualised for a replay clip in Fig. 5b.

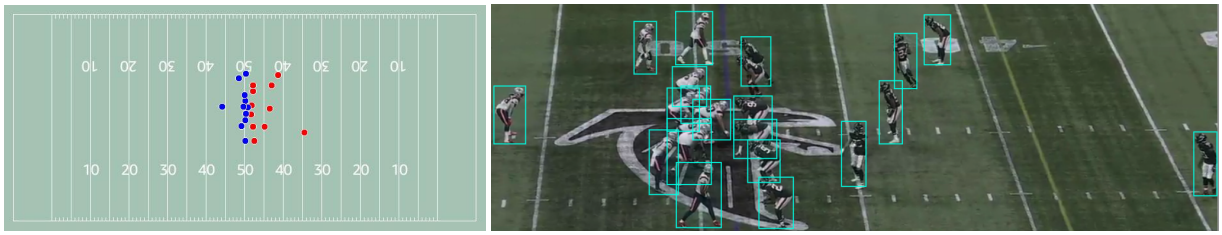
Each point combination is matched with all permutations of 4 players from the pruned set of player detections in the frame (${}^{10}C_4 {}^{8}P_4$ = 352800 possible combinations). The mid-point of the base of each player bounding box is used as the player position in the video frame since even though the RFID sensors are in the shoulder pads, the NGS data is 2D in nature and the player positions are specified with respect to the 2D field plane. As we are using homography mapping between two planes, we need the position of the player on the ground plane of the field. This is also why we cannot use helmet bounding boxes for the homography computation.

The homography matrix is computed for each set of corresponding points along with a cost metric which is the sum of distances to the nearest player location for each mapped point. This creates a one-to-one mapping between every bounding box detected in the video frame and the point coordinates which are within the frame limits. The transformation with the least cost is selected. To make the process efficient, the computation of the 352800 different homography matrices are performed in parallel. Fig. 5c shows the distance between mapped points and bounding boxes which is used for the calculation of the cost function used to select the optimal transformation. The cost function is described in more detail later in the Results section (4.1).

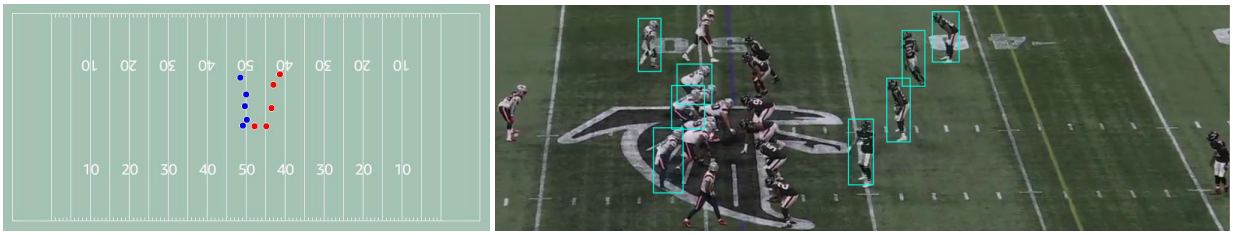
Experiments showed that the homography computed using intelligent sampling of correspondences as opposed to all possible correspondence combinations didn't show any loss of accuracy for the end-to-end system. It was observed that the top few homography matrices (with the lowest cost metric) are similar both in terms of the cost metric value and the homography parameters which validated the hypothesis that a smart selection of the player bounding boxes and NGS points would preserve method robustness.

3.4. Helmet Track Identification

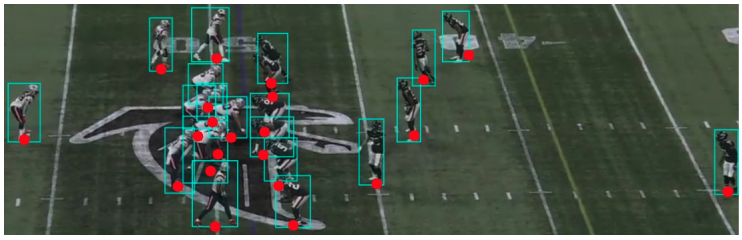
Once the computed transformation maps all NGS 2D plane points to the frame coordinates, this module identifies the corresponding helmet tracks for each mapped point. To track players across frames, we use helmet tracking because helmet bounding boxes are small in size and have a much lower probability of occlusion compared to the player boxes. Helmet tracking gives longer and more pre-



(a) The NGS 2D map and the player detection bounding boxes for the frame of snap for a replay clip.



(b) Selection of 10 player positions from the NGS 2D plane by removal of positions on extreme ends to avoid players which can be out of frame and removal of the most central players who are too close to each other. Selection of 8 player detection bounding boxes with a similar process.



(c) After homography is computed for a large number of combinations from the filtered set of positions and bounding boxes, the best homography is selected based on a cost function which sums up the distance between the mapped homography points(in red) and their nearest bounding box.

Figure 5. Steps followed in the proposed homography computation module which finds the parameters for the homography matrix.

cise tracks as compared to player tracking. The NGS point to bounding box mapping done while computing the cost metric for the homography matrix can sometimes be noisy when the players are congested in one part of the frame.

To ensure a more robust matching, the proposed method creates a dummy player bounding box for the point corresponding to the player of interest with the height and width as the max height and width observed in the player detection results for the frame. We calculate the IoU (Intersection over Union) of this dummy box with every detected box and the player bounding box with max IoU is selected if it has an IoU ≥ 0.25. In the absence of a matching box, the dummy box is assumed to be the desired bounding box. The helmet track for which the bounding box has the maximum IoU with the upper one-third of the identified player bounding box is selected as the helmet track for the target player.

3.5. Homography Tracking

Two methods are proposed to propagate the homography between the NGS plane and the snap frame in the video across multiple frames. The first approach uses the player tracking information generated from the RFID tags, and the second approach uses keypoints detected on the ground plane of the video frame. Homography between frame F_n (frame in NGS data) and F_v (video frame) is used to identify the homography between $F_n + i$ and $F_v + 3i$ (The NGS data is 10 FPS and input video clips have 30 FPS). Propagation methods would fail in case of a change in the camera shot between two frames of the replay clip. Although changes in camera shots are rare within a play in NFL broadcast, in case it happens, the homography needs to be recomputed using the smart sampling based homography computation.

3.5.1 Using Player and Helmet Tracking

Upon identifying the helmet tracks in a frame(F_v), all helmet tracks are selected which persist for the subsequent 3 frames. The corresponding player bounding boxes in the new frame($F_v + 3$) are extracted for these helmet tracks. The mid-point of the base of the player bounding boxes serves as the points in the ground plane of the new video frame($F_v + 3$) which correspond to player points in the NGS 2D plane. These correspondences are utilized to calculate the homography for the new frame. Three different homography matrices are computed using the standard least squares method which uses all correspondences, and using robust methods(least-median of squares and RANSAC [8]) which can ignore outliers. The best homography matrix is selected using the cost metric defined for smart sampling based homography computation method.

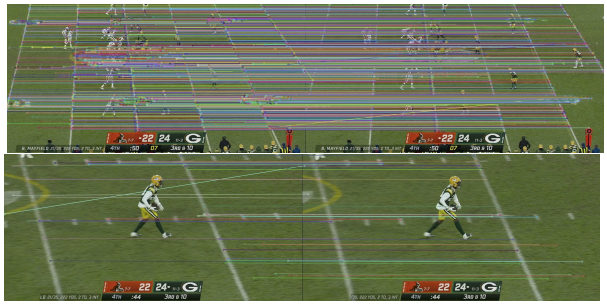


Figure 6. Visualization of the keypoints detected using SIFT [15] and matched in consecutive frames using FLANN [19]. The number of keypoints detected and matched can vary a lot based on the camera angles and portion of the field visible.

The computation fails for cases where less than 4 helmet tracks persist across 3 subsequent frames. This may happen due to motion blur caused by fast camera movement hampering the helmet tracking performance.

3.5.2 Using Ground Plane Keypoints

In this method, key points are identified between subsequent frames using SIFT [15], which are then matched using FLANN [19] to find correspondence between the frames. Keypoints from sections of the frame which do not have any detected players or static graphics are used which ensures that only keypoints from the ground plane are used. Fig 6 shows visualisations of corresponding keypoints found in consecutive frames of a sample clip.

The inverse of the homography for the initial frame is used to transform all the keypoints in the current video frame to points in the NGS 2D plane. The correspondence between the keypoints in the NGS 2D plane and the ground plane of the next video frame is then used to compute the new homography. The robust least-median of squares method is used to compute the transformation to avoid using any incorrect keypoint correspondences. The number of corresponding keypoints identified by this method is usually large and is hence robust.

4. Results

4.1. Homography Computation

We cannot use commonly used evaluation metrics such as IoU, projection error, and reprojection error without ground truth data. In order to assess the quality of a homography mapping and its applicability for consequent helmet identification, we define the following metrics that can be calculated even in the absence of ground truth data.

Total Distance Cost - This is the cost metric referred in section 3.3 and Fig 5c. It sums up the distance in pixels between the mapped homography points and the midpoint of the base of the nearest bounding box. Each bounding box and mapped homography point has a unique match which is ensured by selecting the matches in a greedy way (the lowest distance gets the first match). The matched pair is removed from further rounds of matches to ensure one to one mapping. There might be some noise in this approach to map players but the resulting cost metric is a robust metric to compare the quality of homographies. The bounding boxes at the top and bottom of the frame are ignored because they might contain players who are outside the field and not included in the play.

Median Distance Cost - This is the median distance observed in all the matches made while calculating *Total Distance Cost*. This metric is easier to interpret because the number of matches in the *Total Distance Cost* can vary significantly as it only includes players visible in the video frame.

We evaluated our solution on 151 clips from the 2021 season of NFL. These clips are 25-35 seconds long and cut from the live broadcast of NFL games. It includes the play which is usually 5-10s long and captured from the side camera, some celebration/reactions, and a replay of the event from a different camera angle. The player/helmet detection and tracking are run on the complete clip. After the first snap is identified to synchronise the frames with the NGS data, for homography we focus only on the frames between the start and end of the play. We first use our smart sampling based method to estimate the homography between the NGS data and the predicted snap frame from the video. Subsequently, two methods are used to extend the homography to successive frames until we reach the end of the play. Sometimes the extension might not be possible beyond a certain frame due to lack of availability of corresponding point matches which are used to calculate the extended homographies using methods from sections 3.5.1 and 3.5.2. Table 2 compares the quality of homography obtained using our methods and a state-of-the-art deep learning model for sports field registration by Nie et al. [20].

All our methods significantly outperform the state-of-the-art vision-only model as evident from the drastically low cost values. Another advantage of our methods is that, instead of generating a completely inaccurate homography, it does not compute a transformation in the absence of minimum correspondences required for estimation. Also, the proposed distance cost based metrics introduced in this section are a powerful tool to identify the reliability of an estimated homography which can lead to combining the results from an ensemble of different methods.

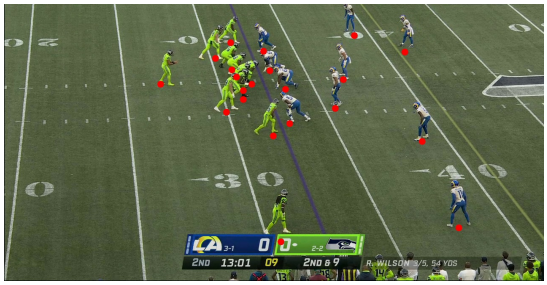
The results summarised in Table 2 use mean and median to summarise results from different clips, it does not capture the internal distribution of metrics for different clips. It is important to understand how many clips each method perform well in. For this, we define ***Good Homography***, which is a homography with *Median Distance Cost* ≤ 30. To un-

Frame	Method	Total Cost		Median Cost		Clips
		Mean	Median	Mean	Median	
0	Ours (3.3)	261.8	229.9	**10.2**	**9.7**	151
	Nie et al. [20]	3094.8	293.4	144.2	12.1	151
1	Track Helmet	293.0	223.8	**9.4**	**9.2**	149
	Keypoints	319.7	251.9	10.7	10.0	148
	Nie et al. [20]	2994.2	313.6	135.5	12.5	151
5	Track Helmet	364.7	260.5	**10.8**	**10.0**	149
	Keypoints	421.6	326.1	12.9	12.3	149
	Nie et al. [20]	3258.3	416.7	153.4	14.5	151
10	Track Helmet	440.5	296.6	**12.2**	**11.8**	147
	Keypoints	513.7	377.2	16.4	15.1	149
	Nie et al. [20]	4497.2	460.4	207.3	16.6	150
20	Track Helmet	844.6	359.5	25.4	13.8	140
	Keypoints	758.3	443.1	**20.5**	**17.0**	148
	Nie et al. [20]	4741.6	550.2	236.1	18.4	149
30	Track Helmet	924.2	469.1	25.7	15.6	86
	Keypoints	758.8	397.8	**23.5**	**18.8**	143
	Nie et al. [20]	4907.5	902.0	291.3	21.5	145
40	Track Helmet	1308.7	639.2	66.1	17.1	37
	Keypoints	939.6	393.1	**45.1**	**29.1**	128
	Nie et al. [20]	3598.6	1975.7	296.5	147.7	131
50	Track Helmet	933.2	571.7	**39.3**	**22.4**	15
	Keypoints	1421.8	694.4	143.3	73.3	108
	Nie et al. [20]	2674.5	1590.0	249.1	157.3	112
75	Track Helmet	373.8	373.8	26.0	26.0	1
	Keypoints	4129.0	2269.3	**565.3**	**238.4**	41
	Nie et al. [20]	2728.1	2774.9	318.3	220.2	43
100	Keypoints	6639.4	5100.8	1437.9	470.4	12
	Nie et al. [20]	2201.7	814.4	**213.8**	**121.9**	16

Table 2. Comparison of *Total Distance Cost* and *Median Distance Cost* (in pixels) for homography estimated using different methods. A lower distance cost is better. The methods are run for the frames within the play from 151 clips and the metrics are obtained for every NGS frame (10 FPS). All clips have different play runtimes. Results for the n^{th} frame of the play are summarised by their mean and median. The number of clips used in the summarised data is mentioned in the "Clips" column. The homography for the first frame (0^{th} frame) is computed using our smart sampling based method. The subsequent frames our tracked using homography tracking methods from section 3.5. All methods have varying clip counts because the homography extension could fail between some frames.

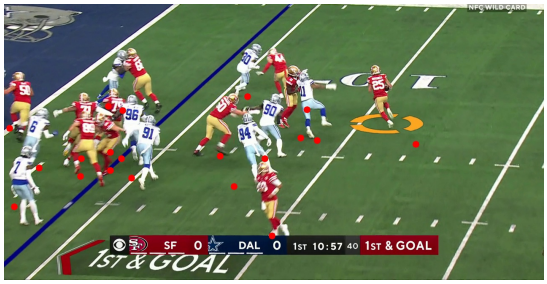
derstand the quality of homographies with different *Median Distance Costs*, we show some examples of mapping generated by different homography in Fig 7. The threshold is selected based on our observation that the helmet track identification approach is robust enough to correctly identify players when the *Median Distance Cost* is less than 30.

Table 3 compares the number of clips for which different methods estimate a *Good Homography*. The results demonstrate the robustness of the smart sampling based method, which estimates a *Good Homography* for the snap frame of every clip. In contrast, the homography tracking methods exhibit limited success in tracking the homography over time, as they are unable to maintain accurate estimates beyond 5 seconds. This is attributed to the dependence on the quality of the homography estimated for the previous frame, which makes it challenging for the tracking methods



(a) *Median Distance Cost* = 6.

(b) *Median Distance Cost* = 15.



(c) *Median Distance Cost* = 31.

(d) *Median Distance Cost* = 119.

Figure 7. Examples of homography mapping with different *Median Distance Cost*.

to recover from bad homography estimates. There is propagation of errors and thus it might need re-initialisation with the smart sampling based approach which is more robust but has higher latency.

4.2. Helmet Track Identification

To test the accuracy of helmet track identification, the proposed workflow visualised in Fig 3 is used to identify the last ball carrier for that particular play in the snap frame of a replay clip. Table 4 captures the results for 160 replay clips. Our smart sampling based homography computation

Frame	Method	Good %	Success %	Computed	Total
0	Ours (3.3)	**100**	**100**	151	151
	Nie et al. [20]	75.5	75.5	151	
1	Track Helmet	99.3	98.0	149	151
	Keypoints	**100**	**98.7**	149	
	Nie et al. [20]	76.8	76.8	151	
5	Track Helmet	99.3	98.0	149	151
	Keypoints	**100**	**98.7**	149	
	Nie et al. [20]	76.2	76.2	151	
10	Track Helmet	98.47	96.0	147	151
	Keypoints	96.0	94.7	149	
	Nie et al. [20]	76.7	76.2	150	
20	Track Helmet	89.3	83.3	140	150
	Keypoints	87.2	86.0	148	
	Nie et al. [20]	69.1	68.7	149	
30	Track Helmet	82.6	49.0	86	145
	Keypoints	79.7	78.6	143	
	Nie et al. [20]	59.3	59.3	145	
40	Track Helmet	59.5	16.8	37	131
	Keypoints	52.3	51.1	128	
	Nie et al. [20]	36.6	36.6	131	
50	Track Helmet	66.7	8.9	15	112
	Keypoints	14.8	14.3	108	
	Nie et al. [20]	13.5	13.4	112	
75	Track Helmet	100	2.3	1	44
	Keypoints	2.4	2.3	41	
	Nie et al. [20]	0	0	43	
100	Keypoints	0	0	12	16
	Nie et al. [20]	**6.3**	**6.3**	16	

Table 3. Comparison of the quality of homography estimated with different methods. The methods are run for the frames within the play from 151 clips and the metrics are obtained for every NGS frame(10 FPS). The number of clips for which the method could generate a homography is mentioned in the "Computed" column. Results for n^{th} frame of the play are summarised in columns "Good %" and "Success %". "Good %" is the percentage of *Good Homography* in the set of all computed homography. "Success%" is the percentage of all clips for which a method estimates a *Good Homography*.

approach gives a good homography (median distance cost $\leq$ 30) for all clips. The helmets identified using the homography and the dummy bounding boxes based approach is also very accurate($\geq 96\%$). The system is robust to snap detection results being off by a few frames. The few failures observed are when the homography mapped points are not exactly on the player's feet and the player of interest is surrounded by a crowd. The loose mapping can be due to the detected snap frame being inaccurate or the player bounding boxes being noisy.

Category	Clips	Percentage
Total Clips	160	
Incorrect Identification	6	3.75%
Correct Identification	154	**96.25%**
No helmet track matched	1	0.625%
Wrong player selected	5	3.125%
Bad Homography	0	0%

Table 4. Accuracy of helmet track identified in the snap frame of the video. The second half of the table gives a breakdown of the cause of incorrect detection of helmet track.

Module	Latency	Compute
Player Detection	25s	Tesla A10 GPU
Helmet Tracking	38s	Tesla A10 GPU
Event(Snap) Detection	16s	3 Tesla A10
Homography Computation	16s	64 vCPUs
Helmet Track Identification	1s	

Table 5. Estimated latency for individual steps involved in the identification of players in the snap frame. The first three modules run in parallel, thus the total end-to-end latency is less than 1 minute (without homography tracking). These estimates are for a replay clip which is 20 seconds long and includes some runtime before and after the actual play.

Module	Latency
Homography Tracking - Track Helmet (per NGS frame)	0.01s
Homography Tracking - Keypoints (per NGS frame)	1s

Table 6. Estimated latency for using homography between NGS plane and video frame m to obtain homography between NGS plane and video frame m+3.

4.3. Latency

The latency for the individual modules are covered in Table 5. The expected total end-to-end latency for identifying a player in snap frame is less than 1 minute. Even though a latency of 1 minute is not low enough to add graphic overlays in live broadcast, it can be used to enhance replay clips which are available to customers on demand when live streaming on OTT services. This only requires computing the homography for one frame. To extend the homography to other frames in the play, the proposed homography tracking methods can be used. The latency for homography tracking/extension is covered in Table 6.

5. Conclusion and Future Work

We present a simple yet effective method to perform highly accurate player identification using homography estimation for sports field registration. The method takes advantage of the RFID based player positions data gathered via wearable RFID sensors. Based on our experiments on a dataset of over 150 American football replay clips, the method outperforms a state-of-the-art deep learning method (which only uses video frames) both in terms of accuracy and success rate in computing the correct homography. These method can be used to automatically generate graphic overlays to identify key players in replay clips which enhances the viewing experience for fans within minutes of the actual play. One of the biggest advantage of the method is that it does not require any annotated data or training for the sports field registration problem. On the other hand, our approach can be used for the automated curation of training data for sports field registration models. Our robust approach can accurately identify key players for more than 96% of replay clips it was tested on.

References

[1] Jacob Andreesen. On-screen graphics and their impact on sports. Available at https://illumin.usc.edu/on-screen-graphics-and-their-impact-on-sports/(2019/04/09), year = 2019. 1

[2] Marco Bertini, Alberto Del Bimbo, and Walter Nunziati. Player identification in soccer videos. In *Proceedings of the 7th ACM SIGMM International Workshop on Multimedia Information Retrieval*, MIR '05, page 25–32, New York, NY, USA, 2005. Association for Computing Machinery. 2

[3] David L. Carey, Tim Bedin, Karl Jackson, and Stuart Morgan. Combining wearable tracking data and deep learning for moving camera calibration. In Arnold Baca, Juliana Exel, Martin Lames, Nic James, and Nimai Parmar, editors, *Proceedings of the 9th International Performance Analysis Workshop and Conference & 5th IACSS Conference*, pages 109–117, Cham, 2022. Springer International Publishing. 2

[4] Yen-Jui Chu, Jheng-Wei Su, Kai-Wen Hsiao, Chi-Yu Lien, Shu-Ho Fan, Min-Chun Hu, Ruen-Rone Lee, Chih-Yuan Yao, and Hung-Kuo Chu. Sports field registration via keypoints-aware label condition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 3523–3530, June 2022. 2

[5] Leonardo Citraro, Pablo Márquez-Neila, Stefano Savarè, Vivek Jayaram, Charles Dubout, Félix Renaut, Andrés Hasfura, Horesh Ben Shitrit, and Pascal Fua. Real-time camera pose estimation for sports fields. *Machine Vision and Applications*, 31(3):16, 2020. 2

[6] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Deep image homography estimation, 2016. 2

[7] Dirk Farin, Susanne Krabbe, Peter H. N. de With, and Wolfgang Effelsberg. Robust camera calibration for sport videos using court models. In *IS&T/SPIE Electronic Imaging*, 2003. 2

[8] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, jun 1981. 2, 5

[9] Ankur Gupta, James J. Little, and Robert J. Woodham. Using line and ellipse features for rectification of broadcast hockey video. In *2011 Canadian Conference on Computer and Robot Vision*, pages 32–39, 2011. 2

[10] Namdar Homayounfar, Sanja Fidler, and Raquel Urtasun. Sports field localization via deep structured models. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4012–4020, 2017. 2

[11] Paul VC Hough. Method and means for recognizing complex patterns, Dec. 18 1962. US Patent 3,069,654. 2

[12] Hyunwoo Kim and Ki Sang Hong. Soccer video mosaicing using self-calibration and line tracking. In *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, volume 1, pages 592–595 vol.1, 2000. 2

[13] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019. 4

[14] Hongshan Liu, Colin Aderon, Noah Wagon, Huapu Liu, Steven MacCall, and Yu Gan. Deep learning-based automatic player identification and logging in american football videos, 2022. 2

[15] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 2, 6

[16] Chun-Wei Lu, Chih-Yang Lin, Chao-Yong Hsu, Ming-Fang Weng, Li-Wei Kang, and Hong-yuan Liao. Identification and tracking of players in sport videos. pages 113–116, 08 2013. 2

[17] Wei-Lwun Lu, Jo-Anne Ting, James J. Little, and Kevin P. Murphy. Learning to track and identify players from broadcast sports videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1704–1716, 2013. 2

[18] Zahid Mahmood, Tauseef Ali, Shahid Khattak, Laiq Hasan, and Samee U. Khan. Automatic player detection and identification for sports entertainment applications. *Pattern Analysis and Applications*, 18(4):971–982, 2015. 2

[19] Marius Muja and David Lowe. Flann-fast library for approximate nearest neighbors user manual. *Computer Science Department, University of British Columbia, Vancouver, BC, Canada*, 5, 2009. 6

[20] Xiaohan Nie, Shixing Chen, and Raffay Hamid. A robust and efficient framework for sports-field registration. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1935–1943, 2021. 2, 6, 7, 8

[21] Kenji Okuma, J. Little, and David G. Lowe. Automatic rectification of long image sequences. 2003. 2

[22] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: an efficient alternative to sift or surf. pages 2564–2571, 11 2011. 2

[23] Arda Senocak, Tae-Hyun Oh, Junsik Kim, and In So Kweon. Part-based player identification using deep convolutional representation and multi-scale pooling. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1813–18137, 2018. 2

[24] Feng Shi, Paul Marchwica, Juan Camilo Gamboa Higuera, Michael Jamieson, Mehrsan Javan, and Parthipan Siva. Self-supervised shape alignment for sports field registration. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 287–296, January 2022. 2

[25] Kanav Vats, Pascale Walters, Mehrnaz Fani, David A. Clausi, and John S. Zelek. Player tracking and identification in ice hockey. *Expert Systems with Applications*, 213:119250, 2023. 2

[26] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Scaled-YOLOv4: Scaling cross stage partial network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13029–13038, June 2021. 3

[27] Fei Wang, Lifeng Sun, Bo Yang, and Shiqiang Yang. Fast arc detection algorithm for play field registration in soccer video mining. In *2006 IEEE International Conference on Systems, Man and Cybernetics*, volume 6, pages 4932–4936, 2006. 2

[28] T. Watanabe, M. Haseyama, and H. Kitajima. A soccer field tracking method with wire frame model from tv images. In

2004 International Conference on Image Processing, 2004. ICIP '04., volume 3, pages 1633–1636 Vol. 3, 2004. 2

[29] Nicolai Wojke and Alex Bewley. Deep cosine metric learning for person re-identification. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 748–756. IEEE, 2018. 3

[30] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3645–3649. IEEE, 2017. 3

[31] Ruiheng Zhang, Lingxiang Wu, Yukun Yang, Wanneng Wu, Yueqiang Chen, and Min Xu. Multi-camera multi-player tracking with deep player identification in sports video. *Pattern Recognition*, 102:107260, 2020. 2