Localisation for Augmented Reality at Sport Events

Patrick Skinner
Department of Computer Science
University of Otago
Dunedin, New Zealand
patrick.skinner@postgrad.otago.ac.nz

Stefanie Zollmann
Department of Computer Science
University of Otago
Dunedin, New Zealand
stefanie@cs.otago.ac.nz

***Abstract*—Sports broadcasts often use pitch aligned graphics to provide additional information to the viewer. This is often achieved by using professional cameras equipped with high accuracy sensors or elaborate manual calibration techniques to measure the broadcasting cameras' position and orientation, allowing the graphics to be accurately matched to the camera view. While previous research has investigated how the camera position and orientation can be estimated for professional broadcast cameras alone, none of the previous works have targeted smartphones.**

In this paper, we investigate whether line pitch markings in combination with feature matching computer vision techniques can be used to estimate an on-site users position and orientation with sufficient accuracy to align augmented reality content with the pitch.

I. Introduction

The goal of our research is to create a method of automatic camera localisation in relation to a sports pitch, allowing us to estimate a camera pose to use when displaying Augmented Reality (AR) content on the pitch. Our method is designed to be run on a standard mobile phone and to be able to integrate into any existing AR framework.

While there have been numerous papers exploring the problem of estimating camera poses at live sports events for professional broadcasting cameras, none of these previous works have devised a solution targeted to mobile devices, allowing spectators in the stadium to automatically initialise an AR experience. The goal of our localization approach is to allow a spectator to use an in-stadium AR application with minimal user input required for the initial calibration and localisation.

Our implementation uses the pitch markings on the field as image features that we can match to a 2D pitch model as seen in Figure 1. Using these lines as feature correspondences allows us to estimate a homography that can be used to align our AR content. Finding likely 2D point correspondences between images is the most common way of computing a homography between images for stereo reconstruction or image registration. However, as the line markings on a sports pitch are the most obvious features, they are likely to give us better results than traditional feature detection methods such as SIFT [1]. We adapt the Direct Linear Transform (DLT) algorithm for computing a homography from point correspondences with a modified version proposed by Dubrofsky and Woodham that allows the use of corresponding line pairs [2].

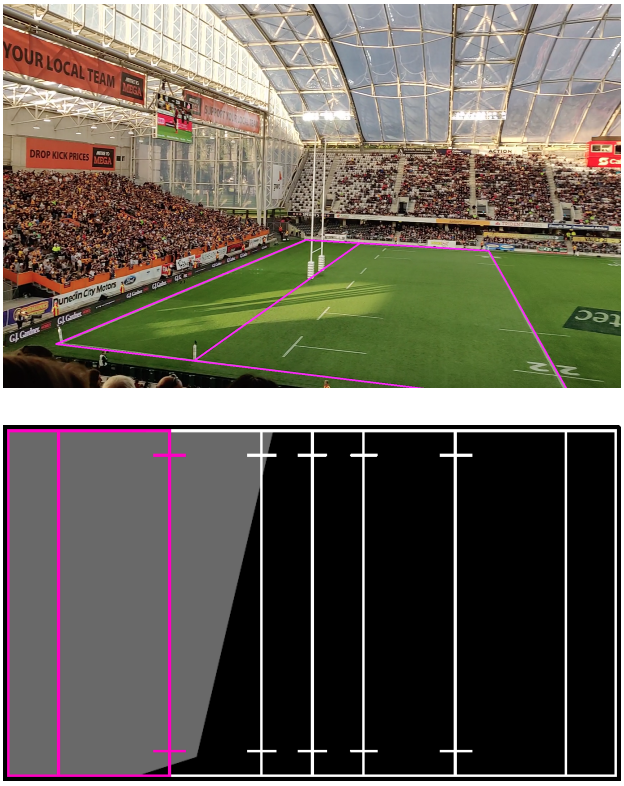

Fig. 1: Matched lines and camera view visualised over our 2D template. Top) Camera view with matched lines overlaid. Bottom) Sports field line template.

Our goal is to combine this modified DLT algorithm with Hough transform based line detection and clustering techniques, finding the homography between the detected pitch markings and a 2D geometric model of the pitch (Figure 1), and to research whether this can be done with sufficient speed and accuracy to be applied to a real-time AR system.

II. Related Work

There have been numerous studies that have explored the process of automatic camera localisation in relation to a sports field. These papers have relied on varying amounts of pre-existing information about the environment, with either

978-1-7281-4187-9/19/$31.0©2019 IEEE

precise information about the location of the camera, or training/calibration footage captured beforehand.

Some of these papers have explored computer vision based approaches, similar to our approach. However many of these are designed around television broadcast cameras with a fixed position that is known in advance, calculating only tilt, pitch and zoom, rather than the arbitrarily positioned and orientated mobile devices we are targeting with our application.

Other papers explore the use of deep learning and neural networks. These require large training sets to be manually annotated beforehand, and these training sets are likely to be unique to a given stadium.

A. Vision Based Approaches

The method proposed by Graham Thomas for soccer pitch localisation [3] requires an initial calibration process to be performed beforehand, to provide an initial rough estimate of the camera position. This process uses multiple images from the same camera at a wide range of tilt and pan angles from previous match footage from the same venue. With this estimate, line detection and a spatialised Hough transform are used to find an initial camera pose for the first frame of footage, on each subsequent frame the lines of the pitch model are projected onto the image and then fit to the detected lines for track the change in pose. This means the overall process is split into three steps, the initial estimation, the initialisation, and the tracking process.

The tracking process runs once per frame and uses the pose estimate from the previous frame to predict the line position. An iterative minimisation process is used to best fit the projected pitch model to observed lines in the image. To find the lines in the image a filter is applied to blue component of the image, to best distinguish between the lines and the surrounding grass, a chroma-keyer is then used to filter the grass from the image. A least squares optimisation is used to compute the tilt, pan and zoom that minimises the sum of squared distance between the detected lines and the lines in the pitch model.

Using 20 images the initial estimation was found to have an uncertainty of +/-15cm with the camera at 90m from the goals. The initialisation process was tested using 250 images selected at random, from this set 208 were able to be used to successfully initialise. Many of the images that were not able to be used successfully were close up shots of particular players that did not contain many visible pitch markings. The average time taken by the initialisation process was 0.99 seconds, running on a single core 3.4GHz CPU.

Gupta et al. [4] proposed a method for homography estimation for ice hockey footage, using a combination of point, line and ellipse matching. This is done as an extension of the Direct Linear Transform (DLT) algorithm for point matching. This method requires five key-frames to be manually chosen for the initialisation process. A set of point correspondences between each key-frame and the geometric model of the rink markings are manually chosen to compute the homography for that frame. For each frame SIFT [1] and SFOP [5] features are detected, these feature descriptors are used to provide an initial homography estimate by matching to the nearest key-frame. This key-frame is used to project a geometric model of the rink onto the current frame, the positions of the lines and ellipses in this projection are used to guide the line and ellipse detection methods in the image frame. As there are no direct point matches available between the geometric model and the current frame, point matches are back-projected from the closest key-frame onto the model to provide a set of matches.

Puwein et al. [6] make use of SIFT [1] and MSER [7] features to compute homographies between cameras in broadcast ice hockey footage. This implementation requires the user to manually calibrate an initial key-frame by selecting the corners of the pitch, providing an initial homography estimate. Once calibrated this method was found to successfully initialise random frames roughly 50% of the time, the failure cases include large zooms and fast movements, as well as frames from cameras level with the playing field.

Our application shares some similarities to these approaches, implementing the modified DLT algorithm to compute a homography from line matches between a source image and a geometric model of the pitch. These methods rely on either manual initialisation processes, or automatic initialisation processes that require camera footage to be processed in advance. Neither of these are applicable to our application, as it is designed to be deployed on a range of mobile devices and initialise from a wide variety of positions, neither which will be known in advance.

Our real world testing has found manual initialisation unsuited to mobile devices. Our test application relied on the user to manually select the four corners of the pitch to provide an initial homography estimate. We found the nature of touchscreen input and the difficulty of holding the device sufficiently still to make this task difficult to complete with the required accuracy.

Our application is designed to work with a minimal amount of pre-existing information, using only a 2D template of the pitch and rough initial estimate of position to account for the fact the pitch is symmetrical on both axes.

B. Deep learning approaches

There are more recent methods that rely on on a machine learning approach for localizing cameras. The method proposed by Homayounfar et al. [8] parameterises the alignment problem using the vanishing points, then uses energy minimization in a Markov random field to encourage agreement between the model and the image. By using the vanishing points to represent this plane the number of degrees of freedom can be reduced, simplifying the optimization problem. Four rays that correspond to the outer lines of the field are found and then used to find the vanishing points, providing the precise location of the field as a 2D rectangle in 3D space.

Unlike the feature and edge detection methods used in previous examples, semantic segmentation is used to find the different field markings. This is achieved using the deep convolutional network VGGNet. This segments the image

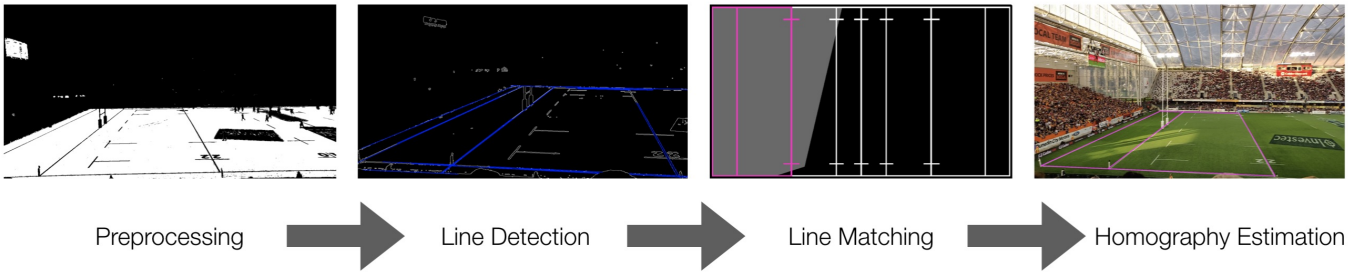

Fig. 2: Overview of our method. There are four main steps within our approach, 1) Chroma Keying, 2) Line detection, 3) Line Clustering and 4) Pose Estimation.

into grass, crowd, and the various different field markings depending on sport.

While this approach produces good results, it requires large training datasets. For this purpose, a large set of frames from broadcast soccer games was manually annotated with ground truth fields. This was used for both training and test data. This requirement for a large amount of training footage is more suited to broadcast cameras, where large quantities of match footage are easily available.

C. Comparisons of localisation methods

Sharma et al. propose and contrast several methods for top view registration of American football video [9]. This includes feature matching approaches using Histogram of Gradients (HOG) features, Chamfer matching, and convolutional neural network based feature detection.

A dictionary of ”camera position to projective transform pairs” is created, where a synthetic projection of the pitch markings is computed for each camera pose. This is done using a ”semi-supervised” approach for generation, where for each training image the user selects four points to compute a homography with, allowing for the generation of a synthetic edge map to match the camera pose. When the feature detection is applied to an image a nearest neighbour approach is used to search the dictionary for the most appropriate pair as a basis for a edge matching procedure.

These methods were tested on both real broadcast footage and a synthetic data set to compare their performance. While the HOG and neural network approaches performed similarly on the synthetic data set, both outperforming the Chamfer matching method, the performance of the CNN was significantly worse than the other methods when applied to real broadcast footage. This suggests that a CNN trained on a synthetic data set is susceptible to the noise found in real world footage. A CNN based approach may perform better when trained using manually annotated broadcast footage. However can be an elaborate task, in particular when large training data sets are required.

Of the feature matching approaches explored by Sharma et al.’s work, the Histogram of Gradients based approach was found the provide the best results on both the synthetic and real world data sets, giving good performance even in the presence of strong shadows, motion blur and varying zoom levels.

As this paper only explores registration for a top down view of the field it is unknown how these methods can be used for sufficiently accurate registration of AR content with the pitch. Again this method requires the use of prerecorded footage from the stadium to generate the dictionary of position-projection pairs.

III. Our Approach

Our method consists of four main steps, 1) Chroma Keying, 2) Line detection, 3) Line Clustering and 4) Pose Estimation (Figure 2). Similar to the existing vision-based approaches, our method involves using detected pitch lines in our image to align a 2D template with the image, allowing us to calculate where the camera is positioned relative to the pitch. We use a probabilistic Hough transform to detect line segments in our input image, and after clustering similar lines and discarding outliers we match these back to our 2D template to create a set of feature correspondences.

A. Preprocessing

Before we can apply this line detection function to our image we need to take several preprocessing steps to transform the input image and to ensure the best results.

The first step is to mask out all parts of the image that are not part of the pitch, such as the crowd and surrounding stadium structure, to ensure that our detected lines are likely to be the pitch markings. This is done using a simple chroma-keying to mask out all non-green pixels. First we convert our RGB image to a HSV (hue, saturation, value) representation, allowing us to use a threshold operation to keep only the areas that are within a chosen range of hues. We chose this threshold to select all pixels that are sufficiently green. These pixels are then used to create a binary image of field, where grass is represented as the white area, and every other pixel is black (Figure 3). This removes the area surrounding the pitch as well as segmenting all the pitch markings.

Before this binary image can then be used as input for the standard Hough line detection process we need to apply an edge detection filter. This first involves applying a Gaussian blur to the input image, followed by the application of the Canny edge detection filter. The output of the canny filter is is another binary representation of the input image where all detected edges are shown as one pixel wide white lines.

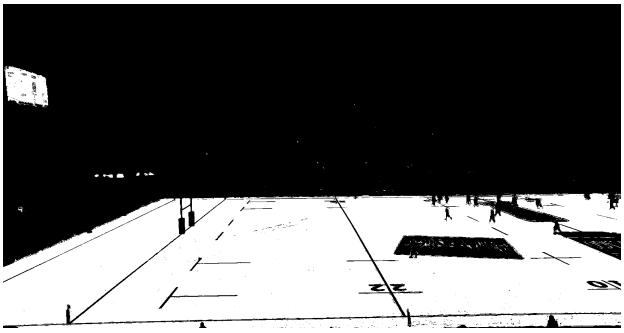

Fig. 3: The output of our chroma-keying step.

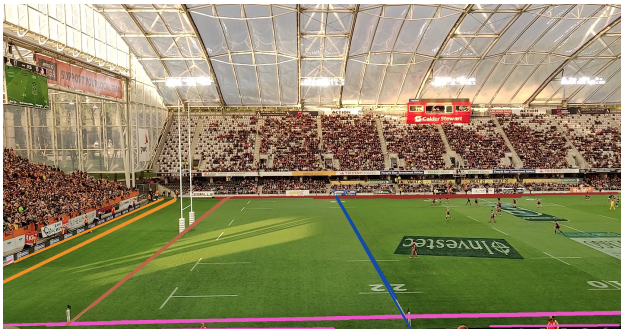

Fig. 5: Line clustering shown by color.

This is the input expected by OpenCV's Hough line detection functions. This function is used to detect all line segments in the image that meet a given criteria, with an acceptable minimum line length and maximum gap between pixels in a line provided as parameters.

B. Line Detection

Once we have completed our pre-processing step we use a Hough line detector in order to detect the pitch lines (Figure 4). The output of the line detector is too noisy for immediate use in our line matching algorithm, with multiple line segments detected for each line on the pitch and a variety of outlier lines that do not belong to any real world pitch marking.

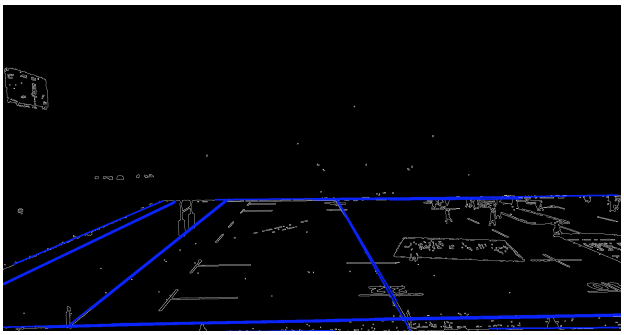

Fig. 4: The output of our line detector.

We have devised a line clustering algorithm to group line segments that are likely to be part of the same pitch marking (Figure 5). This is done by labelling all detected line segments by their angle, as line segments with similar angles are likely part of the same real world line. By grouping lines within a defined number of degrees t_a of each other we effectively separate all the vertical pitch lines in the image into suitable clusters. After initial experiments a threshold, t_a, of 8 degrees turned out to be robust. As the horizontal pitch lines are parallel to each other, an extra step is required to separate them into two groups for the top and bottom of the pitch. This is done by finding the vertical mean and clustering the remaining lines by whether they are above or below this point.

Once we have our lines grouped into clusters we want to produce a single line for each cluster that we can use in our line matching algorithm. We produce a line of best fit through the endpoints of each line in a cluster using an M-estimator technique. This is an iterative function that uses a weighted least squares algorithm to find the best fitting line for a point set. This iterative approach also discards any outlier lines within a cluster, reducing the impact of any noise in our line detector output. This does not guarantee that detected outliers will not effect our best fitting line, and if significant these effects will be reflected in the final homography estimation.

C. Line Matching

Once we have produced a single line for each of our clusters we can use this new line set to find correspondences to our 2D template of the pitch markings.

In our current implementation we use a pitch template to align with either the far left or right of the pitch. In the majority of our test footage the center line of the pitch is obscured by two large logos, meaning we are unable to use that line for template matching. This is likely to be the case for any professional rugby game, though the size and position of the branding may vary such that the center line is usable.

Since we know how our pitch template is structured we can avoid treating this as an unstructured feature matching problem. We start by finding a match for each vertical line in our template, moving from left to right. Moving in this order allows us to ensure the matched line in the image is to the right of the line we matched our previous template line to, maintaining structure. A simple angle threshold is used to make sure we do not match our vertical template lines to any horizontal lines in the image.

It is important to note here that we assume that we roughly know on which side of the stadium/sports ground the user is located. The fact the pitch is symmetrical on two axes means that there are multiple homographies that appear to fit the line model correctly to the pitch, but any AR content will be appear from the wrong orientation if the wrong homography is used. We aim to get this information from the user's GPS or to base this on their assigned seating position.

D. Homography Estimation

Once we have a set of correspondences between our detected and template lines we can use the DLT (Direct Linear Transform) algorithm to calculate a homography. The DLT algorithm is commonly used to compute homographies between

sets of matched feature points in images. However, Dubrofsky and Woodham have shown how the DLT algorithm can be adapted to use line pairs as corresponding features instead of feature points [2].

For each line l in our template we assume there is a corresponding line l' in our image. We consider a single point on l, p and a point on l', p'. We know that p is a point on line l only if:

$$l^T p = 0$$

And that the same is true for a point p' on line l':

$$l'^T p' = 0$$

We know there exists a homography between these points, such that $p' = Hp$. We can substitute this back into our previous figure to find:

$$l = H^T l'$$

This is similar to the correspondence have gives rise to the derivation of the DLT algorithm for point correspondences. We can use this to derive a modified version of the DLT algorithm for line correspondences.

We first convert each of our line pairs to homogenous coordinates, $l = (x, y, 1)$ and $l' = (u, v, 1)$. We can then rewrite our previous equation in the form:

$$c \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = H \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Where c is any non zero constant. From each line correspondence we derive the 2x9 matrix A_i such that $A_i H = 0$:

$$A_i = \begin{pmatrix} -u & 0 & ux & -v & 0 & vx & -1 & 0 & x \\ 0 & -u & uy & 0 & -v & vy & 0 & -1 & y \end{pmatrix}$$

We can then stack these matrices vertically to produce the matrix A. We then perform singular value decomposition on A to produce three matrices, U, S and V.

$$A = USV^T$$

The matrix V is the matrix of right singular vectors. As A will always have a rank of 8, the ninth and final column of V can be used to find the nullspace of A. Which gives the nine values that form H, the homography that relates all our line correspondences.

Once we have computed the homography between our pitch template and the pitch lines detected in the image, we can then use this to position our virtual content with the real pitch. This is done using by solving the Perspective-n-Point problem to estimate the camera pose from the four corners of our transformed template (Figure 6) and the four corresponding 3D points in our 3D reference model.

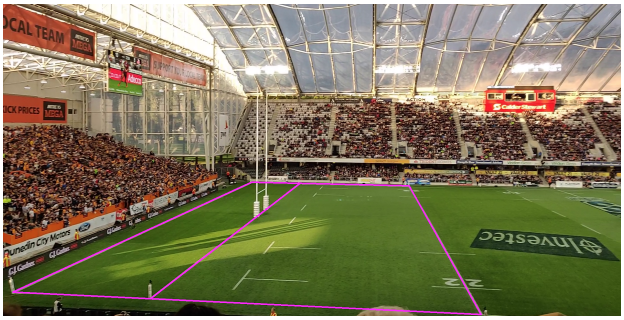

Fig. 6: The 2D template transformed by the our homography.

IV. Results and Performance

From our testing the largest single contribution to the overall runtime of our approach is the initial Hough line detection. While the number of lines detected will affect the runtime of our program, all our real world test footage produces similar enough line counts for this to not be a concern.

Using a random selection of frames from our test footage, we found the average time taken to find a homography was 190 milliseconds, with 60ms of this being the initial reading of the image (Table I).

TABLE I: Overall runtime of our approach.

21 Lines	20 Lines	27 Lines	20 Lines
192ms	205ms	182ms	184ms
196ms	199ms	185ms	187ms
197ms	198ms	189ms	189ms
199ms	200ms	186ms	195ms
196ms	199ms	184ms	189ms

As we initialise from one side of the pitch, the error in angle become more apparent towards the center of the pitch as our projected lines deviate from the real pitch markings (Figure 7, Middle). The effects of these errors on the overall user experience will need to be investigated.

Using six distinct test images that could be successfully initialised, we measured the distance between the four corners of our transformed partial template and our manually chosen ground truth corners. We found the average distance to be 4.83 pixels. With the best fitting corner being 1.4 pixels from the ground truth, and the worst being 16 pixels away. With the resolution of our test images being 1920 by 1080, we believe this is sufficiently accurate to provide a convincing registration of AR content, though this will have to be assessed through a future user study.

Currently most of our failures to initialise are due to the detection of outlier lines, generally at the borders of the field. Figure 7 (Left) shows such a failure case. The border between the stands and pitch has provided more detected Hough lines than the bottom pitch marking, resulting in our line fitting algorithm discarding the pitch lines as outliers.

Discarding detected lines at the very border of field will likely resolve this issue. Fitting a contour around the binary image produced by our chroma-keying step will likely allow us to find where the field borders are.

Fig. 7: Problems of our approach. Left) The bottom border of the field influencing our homography estimation. Middle) Error accumulated from detected outliers. Right) Homography estimation from the center being affected by a logo.

V. Future Work

The current version of our method takes a single input frame and returns a computed homography, each frame is processed independently with no consideration to any of the previous frames or the previously computed homographies.

When applying our method to a live video feed we have the option of using the previously computed homography and resulting line positions to guide our line detector on the following frame. We can reproject our template back onto the field and search for lines close to those from our previous frame. Providing a previously calculated homography will also us to use this as an initial estimate for our homography estimation on the following frame, which should increase the speed of the estimation process for all subsequent frames.

Our future plans include taking our approach for homography estimation and using it in an AR mobile application for real-time sports spectators, combining our pose estimation with the device tracking functionality provided by ARKit. Assuming all our AR content is anchored to a virtual plane, we can use a Perspective-n-Point method to calculate the camera position from the 4 corners of the virtual plane and the 4 corners of our 2D template transformed by the homography, allowing us to align our virtual pitch with the real world pitch.

Using the positional tracking provided by ARKit will allow us to keep content aligned between reinitialisations, as well allowing us to estimate which section of the template we will need to align to for the next reinitialisation.

Future work may include expanding the range of initialisation angles. Currently, we can only initialise using the extreme left or right of the pitch, depending on the user's seating position. The initialisation provides best results when looking at the closest corner of the pitch, as closer pitch markings are more clearly visible and more easily detected.

Initialising from the center of the pitch has its own problems, particularly with logos painted on the pitch obscuring the center line or interfering with the line detector as seen in the figure above (Figure 7, Right). A more robust solution will be required to avoid the influence of these advertisements on our homography estimation.

VI. Conclusion

We have shown that line detection can be used to successfully compute a homography to a limited subsection of a 2D pitch model in an acceptable timeframe.

We believe our method can form the basis of a robust AR application. We have identified the current shortcomings of our method, and common failure cases. Solutions to these problems have been proposed, and will be investigated in our future iterations of our program. We will also undertake performance analysis to compare our method to existing localisation methods, as well as testing across a wider range of environments and sports to test how well our approach can be generalised to a variety of use cases.

Acknowledgments

This project is supported by an MBIE Endeavour Smart Ideas grant (UOOX1705). We thank Animation Research Ltd, Forsyth Barr Stadium, the Highlanders, Otago Rugby (ORFU) and OptaPerform for their support.

References

[1] D. G. Low, "Distinctive image features from scale-invariant keypoints," in *International Journal of Computer Vision*, 2004, pp. 91–110. [Online]. Available: https://www.cs.ubc.ca/ lowe/papers/ijcv04.pdf

[2] E. Dubrofsky and R. J. Woodham, "Combining line and point correspondences for homography estimation," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5359 LNCS, no. PART 2, 2008, pp. 202–213.

[3] G. Thomas, "Real-time camera tracking using sports pitch markings," *Journal of Real-Time Image Processing*, vol. 2, no. 2-3, pp. 117–132, 11 2007.

[4] A. Gupta, J. J. Little, and R. J. Woodham, "Using line and ellipse features for rectification of broadcast hockey video," in *Proceedings - 2011 Canadian Conference on Computer and Robot Vision, CRV 2011*, 2011, pp. 32–39.

[5] W. Förstner, T. Dickscheid, and F. Schindler, "Detecting interpretable and accurate scale-invariant keypoints," in *Proceedings of the IEEE International Conference on Computer Vision*, 2009, pp. 2256–2263.

[6] J. Puwein, R. Ziegler, J. Vogel, and M. Pollefeys, "Robust multi-view camera calibration for wide-baseline camera networks," in *2011 IEEE Workshop on Applications of Computer Vision, WACV 2011*. IEEE, 2011, pp. 321–328.

[7] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," in *Image and Vision Computing*, vol. 22, no. 10 SPEC. ISS. Elsevier Ltd, 9 2004, pp. 761–767.

[8] N. Homayounfar, S. Fidler, and R. Urtasun, "Soccer Field Localization from a Single Image," 4 2016. [Online]. Available: http://arxiv.org/abs/1604.02715

[9] R. A. Sharma, B. Bhat, V. Gandhi, and C. V. Jawahar, "Automated Top View Registration of Broadcast Football Videos," in *Proceedings - 2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018*, vol. 2018-Janua. Institute of Electrical and Electronics Engineers Inc., 5 2018, pp. 305–313.