Research White Paper

WHP 146

January 2007

Real-Time Camera Pose Estimation for Augmenting Sports Scenes

G. A. Thomas

BRITISH BROADCASTING CORPORATION

Real-Time Camera Pose Estimation for Augmenting Sports Scenes

G. A. Thomas

Abstract

A method for computing the position, orientation and focal length of a camera is presented, designed for use in applications such as the real-time overlay of graphics on a football pitch. The method uses markings on the pitch, such as arcs and lines, to compute the camera pose. A novel feature of the method is the use of multiple images to improve the accuracy of the camera position estimate. A means of automatically initialising the tracking process is also presented, which makes use of a modified form of Hough transform.

This paper was presented on 29th November 2006 at the 3rd European Conference on Visual Media Production (CVMP), held at the IET, Savoy Place, London.

Additional key words: camera tracking, pose estimation, football, soccer

White Papers are distributed freely on request.

Authorisation of the Head of Research is required for publication.

Real-Time Camera Pose Estimation for Augmenting Sports Scenes

G.A. Thomas

BBC Research, Kingswood Warren, Tadworth, Surrey, UK
graham.thomas@rd.bbc.co.uk

Keywords: camera tracking, pose estimation, football, soccer

Abstract

A method for computing the position, orientation and focal length of a camera is presented, designed for use in applications such as the real-time overlay of graphics on a football pitch. The method uses markings on the pitch, such as arcs and lines, to compute the camera pose. A novel feature of the method is the use of multiple images to improve the accuracy of the camera position estimate. A means of automatically initialising the tracking process is also presented, which makes use of a modified form of Hough transform.

1 Introduction

In order to present analysis of sports events to TV viewers, a common requirement is to be able to overlay graphics on the image, which appear to be tied to the ground. It is also useful to be able to show markings at the correct absolute scale, such as distances from a player on a football pitch to the goal. This requires knowledge of the camera pose (position and orientation), as well as the focal length, i.e. a full metric camera calibration. The calibration data generally needs to be generated at full video rate (50Hz or 60Hz). An example of some typical overlaid graphics is shown in Figure 1.

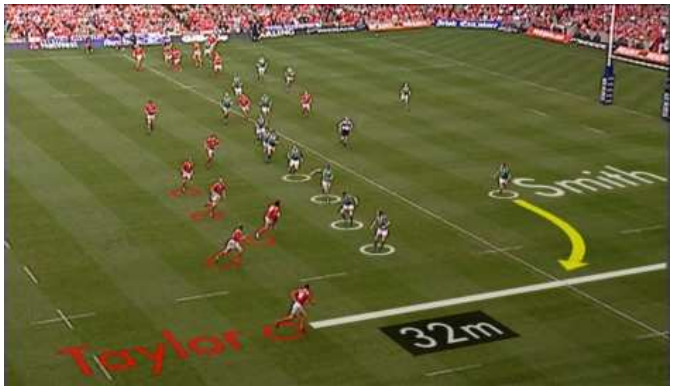

Figure 1 - Graphics overlaid using camera pose data

One way in which camera calibration data can be derived is by performing an initial off-line calibration of the position of the camera mounting using a theodolite or range-finder, and mounting sensors on the camera and the lens to measure the pan, tilt, and zoom. However, this is costly and sometimes very difficult, for example if it is not possible to gain access to the camera mounts, or if cameras are mounted on non-rigid structures such as a crane.

A more attractive way of deriving calibration data is by analysis of the camera image. The lines on a sports pitch are usually in known positions, and these can be used to compute the camera pose. In some sports, such as football, the layout of some pitch markings (such as those around the goal) is fully specified, but the overall dimensions vary between grounds. The Football Association specifies that the pitch length must be in the range 90-120m, and the width 45-90m; for international matches, less variation is allowed (length 100-110m and width 64-75m) [3]. It is thus necessary to obtain a measurement of the actual pitch.

One example of past work in this area is [10], in which an exhaustive search over position, orientation and field-of-view is performed to match a wire-frame model of the pitch lines to the lines detected in the camera image. Although this method requires no manual initialisation, the exhaustive search approach suggested is likely to result in processing times that are too long to be practical for applications in sports graphics generation, where the system needs to initialise in about 1 second. While the quantisation inherent in the searching process in [10] may be sufficient for the application described (gathering statistics on player position), it is unlikely to result in a camera pose that has sufficient accuracy for convincing graphical overlay.

An example of a method that computes the camera pose by iterative minimisation of the error between observed lines and reprojected lines from a model is [9]. This method works by locating edge points in the image close to the predicted edge position, but no method of automatic initialisation is presented. It is stated that the method tracks with a small amount of jitter, which can be reduced by using texture information for tracking in addition to the edge information.

Other researchers have investigated the detection of specific kinds of line features that occur in football. For example, [11] presents a method for detecting the ellipse in the image formed by the centre circle, although this only works when more than half of the circle is visible, and makes no use of other line features in the image. Billboards around the pitch

can also be tracked [1], although many football grounds now have animated or scrolling billboards, which would make such an approach unsuitable.

In this paper we propose a method based on line tracking, similar to [9] in that the camera pose is computed to minimise the reprojection error to observed edge points. We also use a variant of the multi-hypothesis approach that [9] describes: only those edge points closest to the predicted line position are considered, rather than using all points within a given search area. This provides robustness to the appearance of other nearby edge points. However, our method includes an automatic initialisation process, similar in concept to the exhaustive search of [10], but implemented in such a way that the process can be carried out in about one second. We also take advantage of the fact that TV cameras at outside broadcasts are often mounted on fixed pan/tilt heads, so that their position remains roughly constant over time. An overview of the whole process is given in Figure 2.

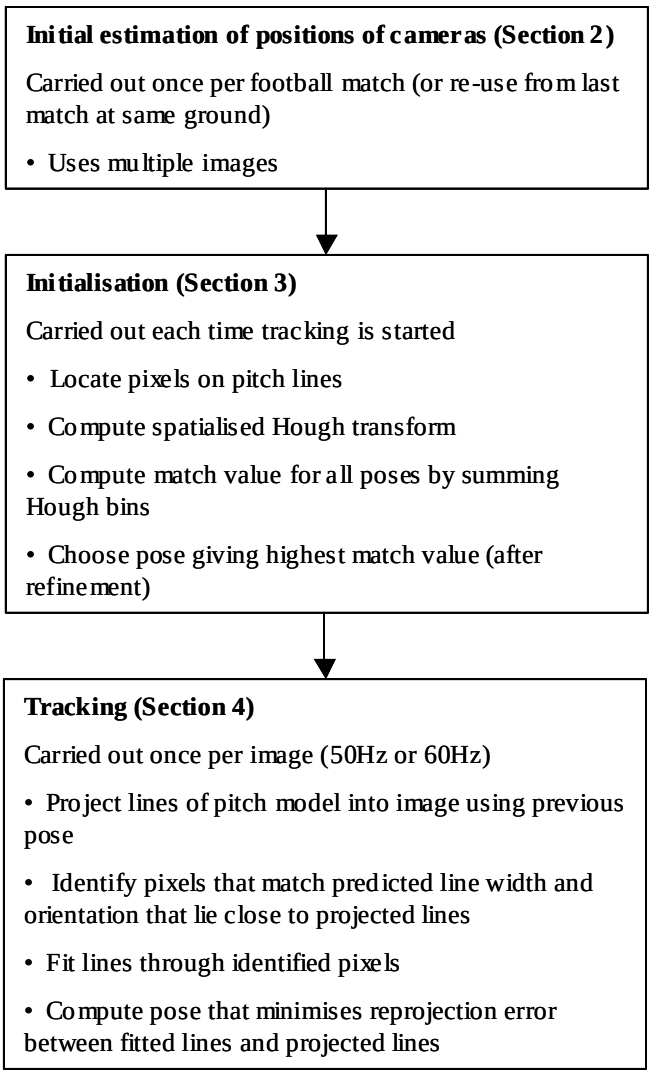

Figure 2 - Overview of the method

The following section describes the method used to estimate the position of each camera mounting, when the system is first set up. Section 3 explains the initialisation method, which is invoked when starting to track, or when the system has lost track (for example, if the camera view moved away from showing any pitch lines). Section 4 discusses the real-time tracking process. Section 5 presents some results, and the remaining sections present discussions and conclusions.

2 Estimation of the camera position

Most cameras covering events such as football generally remain in fixed positions during a match. Indeed, the positions often remain almost unchanged between different matches at the same ground, as the camera mounting points are often rigidly fixed to the stadium structure. Figure 3 shows a typical camera mounting.

Figure 3 - A TV camera at the Twickenham Rugby Stadium

It therefore makes sense to use this prior knowledge to compute an accurate camera position, which is then used as a constraint during the subsequent tracking process. Estimating camera position from image correspondences can be a poorly-constrained problem, particularly if focal length also needs to be estimated. To improve the accuracy, we propose to use multiple images, and solve for a common camera position value.

Constraining the position to be consistent with multiple images reduces the degrees-of-freedom available when computing the pose. This might be expected to lead to a slight increase in the reprojection error, as there are fewer degrees of freedom available. However, if the position computed is close to the true position, then this effect should be negligible, assuming that there are no other effects such as the camera position moving significantly as the camera pans (for example, due to the optical centre of the camera lying a little distance away from the axis about which the camera pans).

The pose computation method described in Section 4 is used to compute the camera position, orientation and field-of-view, for a number of different camera orientations, covering a wide range of pan angles. The pose for all images is computed simultaneously, and the position is constrained to a common value for all the images. This significantly reduces the ambiguity between the distance of the camera from the reference features and the focal length that is inherent in most

camera calibration methods. By including views of features in a wide range of positions (e.g. views of both goal areas), the uncertainty lines along different directions, and solving for a common position allows this uncertainty to be significantly reduced. Examples of the position estimation process are given in Section 5.1.

Cameras usually cannot roll (i.e. rotate about their direction of view). However, this does not necessarily mean that the roll angle can be assumed to be zero. The camera may not necessarily be mounted flat on the pan/tilt head, or the head itself may not be aligned with the pan axis exactly vertical. Also, it is usually assumed that the plane of the pitch is essentially horizontal, but this may not be the case. Each of these effects can give rise to what appears to be a small amount of camera roll, which could vary with pan or tilt, depending on the cause. One solution would be to compute camera roll for every image during the tracking process, but this introduces an additional degree of freedom, and therefore will increase the noise sensitivity and increase the minimum number of features needed for accurate pose computation. Another option would be to attempt to solve for each of these small mis-alignments separately. Instead, we chose to solve for the apparent rotation of the pitch plane about the dominant direction of the camera view. We found that in practice this accounts sufficiently well for the combination of effects described earlier. The pitch rotation is computed during the global position computation process, giving a single value optimised for all the images used.

Although it is generally best to apply this technique by manually selecting around 10-20 images from each camera (for example, from a video tape recorded from a previous game at the ground), we have developed a method whereby images can be acquired automatically to refine the position computation during the tracking process. The camera position and orientation are first set manually to match a given image. The tracking process (Section 4) then computes new values of pan, tilt and roll for every image, using the previous pose as an initial estimate. Each time a pose is computed that has values of pan or tilt that are significantly different from those of images used previously (typically by about 10º), the image is added to the list of images used for global position computation, and a new globally-consistent position and pitch plane rotation are calculated, incorporating the newly-captured image. This allows the system to automatically refine its estimated position during tracking.

The camera position and pitch rotation computed in this way are then used for the initialisation and tracking processes described below.

3 Initialisation

3.1 Approach

The Hough transform [6] is a well-known way of finding lines in an image. It maps a line in the image to a point (or accumulator "bin") in Hough space, where the two axes represent the angle of the line and the shortest distance to the centre of the image. If the camera pose is known roughly, it is possible to predict which peak in Hough space corresponds to which known line in the world, and hence to calibrate the camera. However, if the camera pose is unknown, the correspondence can be difficult to establish, as there may be many possible permutations of correspondences. Furthermore, if some lines are curved rather than straight, they will not give rise to a well-defined peak and are thus hard to identify.

Rather than attempting to establish the correspondence between world lines and peaks in Hough space, we use the Hough transform as a means to allow us to quickly establish a measure of how well the image matches the set of lines that would be expected to be visible from a given pose. A "match value" for a set of lines can be obtained by adding together the set of bins in Hough space that correspond to the lines we are looking for. Thus, to test for the presence of a set of N lines, we only have to add together N values from the Hough transform, rather than examining all the pixels in the image that we would expect the lines to lie upon.

We use this in an exhaustive search process, to establish the match value for each pose that we consider. This provides a much faster way of measuring the degree of match than that used in the exhaustive search process proposed in [10]. For each pre-determined camera position, we search over the full range of plausible values of pan, tilt, and field-of-view, calculating the match value by summing the values in the bins in the Hough transform that correspond to the line positions that would be expected.

By representing a curved line as a series of line segments, curves can also contribute to the match, even if they do not give rise to local maxima in the Hough transform. We used one segment for every 20º of arc. Although specific forms of Hough transform exist for circle or ellipse detection (such as that used in [11]), we chose the line segment approach to allow both curves and lines to be handled in a single process.

The following two sub-sections explain the kind of Hough transform used, and give further implementation details.

3.2 A variant on the Hough transform that maintains spatial information

A point in a conventional Hough transform represents a line of infinite length, i.e. the information about which part of the line a contributing point lies on is lost. This means that, when measuring the degree of match for a line segment from the

value in the corresponding Hough bin, all edge pixels that are co-linear with this segment will be considered, even if they lie beyond the ends of the line segment. This makes the match value less reliable, as noise samples, or samples from other lines, will contribute when they should not.

Various methods of incorporating spatial information in a Hough transform have been proposed before; for example, [5] describes an approach in which the input image is divided into a quad-tree, so that the lines in a particular region can be identified.

We chose a relatively simple approach to retaining spatial information, which we refer to here as a spatialised Hough transform. Rather than sub-dividing the image into 2D regions, we divide each line into S 1D segments: this maintains a common set of bins for the whole image, with each bin being subdivided into S sections. Rather than divide every line into S equal sections, for simplicity we divide the line by reference to either the horizontal portion of the image in which it lies (for lines that are closer to horizontal than vertical), or the vertical portion (for lines that are closer to vertical). Thus to determine the sub-bin that a given pixel contributes to, it is only necessary to examine either its x or y coordinate, depending on the angle that the bin in the transform corresponds to. The resulting transform has three dimensions: distance and angle (as in a conventional Hough transform), and "distance from picture edge", measured from either the bottom or the left of the image, depending on the slope of the line. This third axis has length S, and is in units of picture width divided by S, or picture height divided by S.

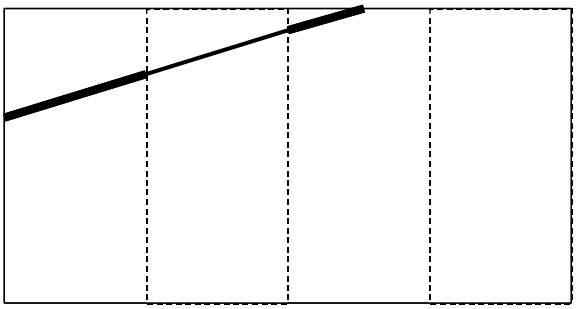

Figure 4 - Dividing a line into multiple segments for the spatialised Hough transform

Not all lines will have positions or angles that need all the bins, as they may not cover the full width or height of the image. However, this approach leads to a very efficient way of computing the sub-bin for any given pixel, and this was considered preferable to a more complicated method that might make slightly more efficient use of memory. Figure 4 shows an image in the case where S=4, and a line closer to horizontal than vertical which is thus divided horizontally. The line is divided into the three segments shown by alternating thick and thin lines; the 4th spatial sub-division of this line will never contain any points, as it would lie above the top of the image.

3.3 Implementation

The steps in our initialisation process are as follows:

Step 1. We first identify all the pixels in the input image that may correspond to the lines we are interested in. The line detection filter used in the tracking process described in Section 4.2 is used. However, this filter assumes the width and orientation of the line is known, which is not the case at this stage. Therefore we apply the filter several times, with a range of assumed line widths, and using both horizontal and vertical orientations. The outputs of these filtering operations are added together. For each pixel where this sum is above a given threshold, we add a value to the appropriate sub-bins of the spatialised Hough transform. The value added is proportional to the summed filter outputs, so that a higher weight is given to more reliable information.

Step 2. For each camera position estimated as described in Section 2, we step through the plausible ranges of pan, tilt and zoom, and project the lines of the pitch model into the image. The step size is chosen to be equivalent to a fixed number of pixels in the image; in the case of zoom this is the motion caused at the edge of the image. For each candidate pose where at least three lines are visible, we compute the list of sub-bins in the spatialised Hough transform that correspond to the projected lines. The contents of all the sub-bins are summed to obtain a weighting value for this pose. We make a list of the 5-10 poses having the highest sums (ignoring poses that are close to other poses having a higher sum, as they are not useful local maxima). It is worth noting that the list of sub-bins to be used for each pose depends only on the estimated camera positions and the geometry of the pitch model, so the entire list of bins for all possible poses can be pre-computed once the camera positions have been determined as described in Section 2. This significantly speeds up the search process, as no projection of pitch lines or mapping of lines to Hough space needs to be carried out during the initialisation process.

Step 3. Each of the high-scoring poses identified in the previous step are used in turn to initialise the tracking process (Section 4). Several iterations of the tracking process are run for each pose, and the pose that returned the highest match value at the end of this process is used as the final pose. The tracking process will "lock on" to lines that are within the search window it uses, and therefore this will generate a more accurate measurement of the match value for each pose than the value obtained by summing the bins in Hough space in Step 2. The search window size used in the tracking process is chosen to be slightly larger than the step size used in the exhaustive search in Step 2, to ensure that the tracking process finds the lines.

4 Tracking

4.1 Approach

The tracking process uses the pose estimate from the previous image, and searches a window of the image centred on each predicted line position for points likely to correspond to pitch lines. A straight line is fitted through each set of points, and an iterative minimisation process is used to minimise the distance in the image between the ends of the observed line and the corresponding line in the model.

4.2 Implementation

The steps in the tracking process are as follows:

Step 1. Each line in the pitch model is projected into the camera image using the previous camera pose as an estimate of the current pose. For lines that are at least partly visible, a window around each projected line is processed to compute a measure of the extent to which each pixel is likely to lie at the centre of a line, using a simple line detection filter that uses knowledge of the predicted line width and orientation.

The line detection filter operates by computing the difference between a pixel and two adjacent pixels, separated by half the width of the line we are looking for. By taking the minimum of these two difference values, the filter detects lines and ignores edges of regions significantly wider than a line. Ideally we would orient the filter so that it operated at right angles to the line we wanted to detect; however, to simplify the processing we apply the filter either horizontally or vertically, depending on whether the line is closer to horizontal or vertical. The assumed width of the line is adjusted to reflect its width in the horizontal or vertical direction, as appropriate. The local maximum of the filter output is taken to identify a pixel at the centre of the line.

The filter is applied to the blue component of the image, as this distinguishes well between the green grass and the white line; alternative methods of deriving a single-component signal from a colour image may be appropriate in applications with other line or background colours. The adjacent pixels must also have a colour in the range expected for grass (this is to discard points in areas of the image such as the crowd or advertising hoardings). A hue-based chroma-keyer is used for this purpose. This line detection filter has the advantage of being very fast, having a simple filter kernel involving only two subtractions and a MIN operator. Figure 5 summarises its operation.

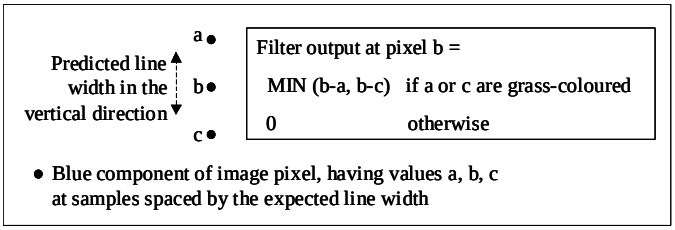

Figure 5 – The line detection filter, illustrated for the case of a near-horizontal line

For each column or row of pixels that the projected line crosses (depending on whether the line is closer to horizontal or vertical), we find the pixel closest to the projected line for which the line detector output is above a given threshold. This provides robustness to the appearance of other nearby edge points, similar to the multi-hypothesis approach in [9].

Step 2. For each line, a straight line is fitted through the set of detected pixels, weighting each point in accordance with the amplitude of the line detector output. An overall weight is assigned to each line, based on the number of points added. The total of these weights gives a match value for this pose that can be used to trigger re-initialisation if it falls below a given threshold; it is also used when this processing is applied to pose selection in Step 3 of the initialisation process (Section 3.3).

Step 3. A least squares iterative optimisation is performed, to compute the pan, tilt and zoom values that minimise the sum of the squares of the distance (in image pixels) from the end points of the fitted lines to the lines in the pitch model projected into the camera image. For the purpose of computing the distance from an observed end-point to the projected pitch model line, the lines in the pitch model are assumed to be of infinite extent; this allows the distance to be computed in a straightforward manner. The minimisation process is similar to the well-known Levenberg-Marquardt method, using the pose from the previous image as the initial estimate. Note that when this process is being applied in the initial position estimation stage, the minimisation process also varies the camera position and pitch orientation, and is applied simultaneously across a number of images (see Section 2).

As an alternative to the line-fitting process in Step 2, the set of edge points from Step 1 can be used directly in Step 3, instead of using just the two end-points of the fitted line. This increases the amount of computation in the minimisation process, as the number of equations to be minimised goes up significantly (from around 16 to about 800 in a typical case of 8 visible lines, each about 100 pixels long).

An advantage of using the detected line points directly is that lens distortion can be estimated. This cannot be done accurately if a straight line is fitted to each line, as information on the curvature of the line is lost. By using the edge points directly, the minimisation process can "see" the way in which adjusting the distortion parameter(s) affects the reprojection error for each point on the line, and thus the

distortion can be estimated. We have found this to be useful in applications where a very accurate calibration is needed, but for graphics overlay applications this is usually not necessary.

5 Results

The results presented in this section were obtained using broadcast video from a football match (Manchester City vs. Arsenal, 25th April 2005). Most of the material was shot from the main camera (positioned roughly in line with the centre line), with other shots from two cameras positioned roughly in line with the two 18-yard lines. Other cameras were occasionally used (for example, positioned behind the goals, or providing close-up shots of players). The material was standard broadcast quality (720x576, 50Hz interlaced, 16:9 aspect ratio, in 4:2:2 YUV format). The processing was applied to individual fields after horizontal subsampling by a factor of two, so each processed image had dimensions 360 x 288. We found that subsampling the image horizontally had a negligible effect on the accuracy of the algorithm, but reduced the processing time by around 35%. An example image from the material is shown in Figure 6.

The coordinate frame used had the origin at the centre of the pitch. The x axis was towards the right goal, the y axis pointed up, and the z axis pointed towards the main camera.

Figure 6 – An image from the test material

5.1 Computation of initial position

To illustrate the benefit of computing the camera position using multiple images, a 20-minute section of the match coverage were used. Images were manually selected, taking images that all came from the main camera, covering most of the typical ranges of pan, tilt and zoom.

Figure 7 shows a plot of the positions in the xz plane computed when each image was processed individually. The predominance of values along two diagonal lines is due to the fact that the majority of the calibration information comes from lines around the two goals, so ambiguity between changes in focal length and distance to the features used for calibration tends to be in a direction from the camera to the two goals. This can be seen more easily in Figure 8, which shows a wider view of the data points in Figure 7, including the pitch markings.

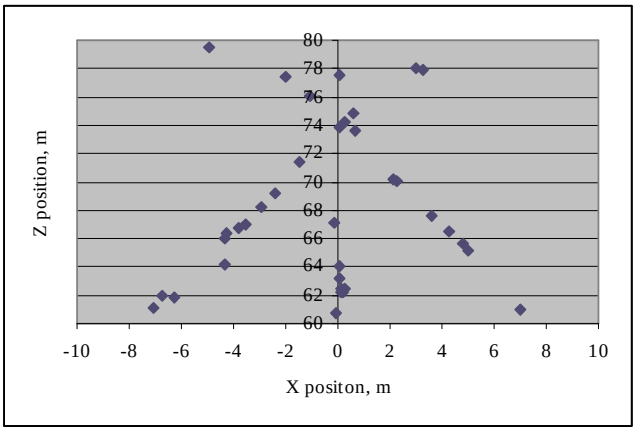

Figure 7 - Position computed using individual images

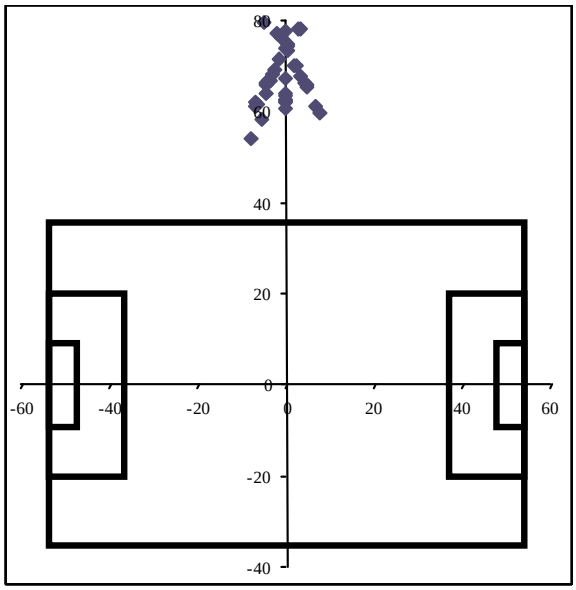

Figure 8 - Estimated camera positions in relation to the whole pitch

The position computation process was then repeated, using data from multiple images simultaneously. Each time an image was selected, the camera pose was recomputed, using both the lines visible in the new image, and all the images captured previously, forcing a common position to be used for all images. The 20-minute section of the match coverage was split into four 5-minute sections, and each was processed separately, to provide an indication of repeatability. The plots in Figure 9 show how the calculated position changed as more images were used.

It can be seen that after about 10-20 images, the computed positions stabilise to within about 0.3m of each other, to a value around (0.2, 18.5, 75.0). The largest uncertainty is in the z direction, which follows from the fact that the camera is on average viewing the scene within 20-30º of the z axis.

It is interesting to compare the estimated position computed in this globally-consistent manner, to the positions in Figure 7 calculated using the individual images. It is worth noting that the globally-optimum position is very close to where lines through the groups of measurements would cross.

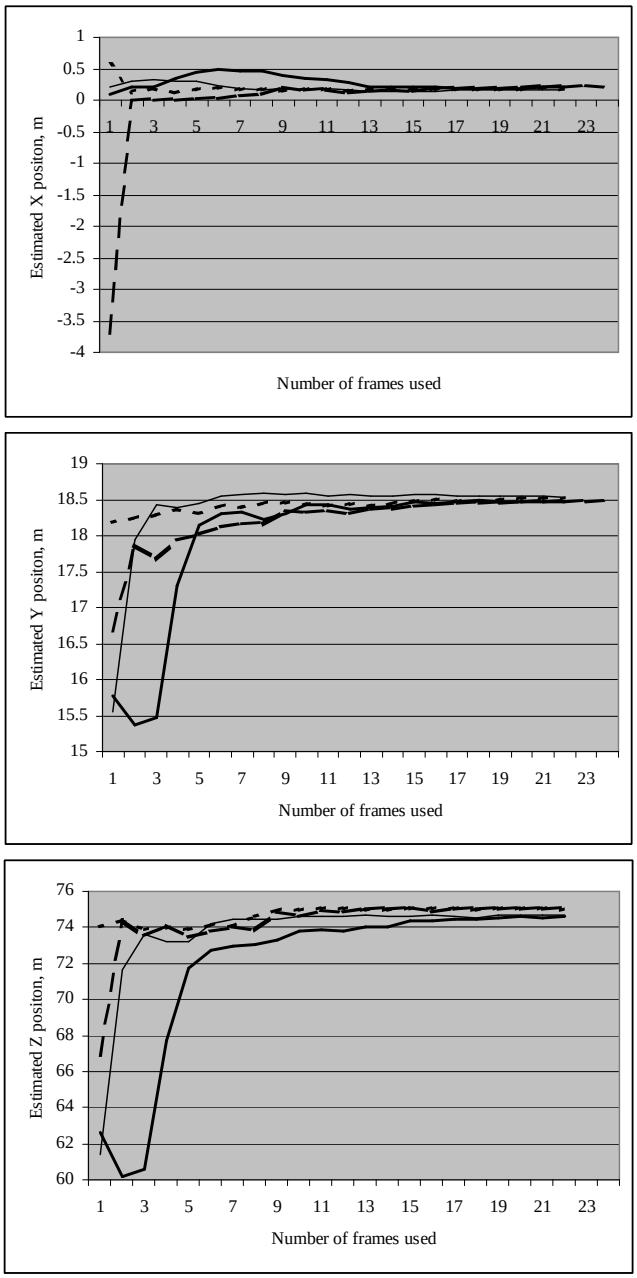

Figure 9 - Refinement of the camera position estimate (x, y and z) by using multiple images, for four sets of images

To test the validity of the assumption that a common position can be used for a wide range of viewing directions, the reprojection error (expressed as an RMS error in pixels between the ends of the measured pitch lines and the modelled lines) was measured after each new image was included in the position computation, and plotted in Figure 10. The error tends to vary in an apparently random way between images; this is due to the varying number of lines visible in each image, and the degree to which effects such as lens distortion and inaccurate positions of the true lines affect each shot. There is no significant increase in error as additional images are processed, confirming that forcing the position to be consistent with all the captured images is not over-constraining the computation.

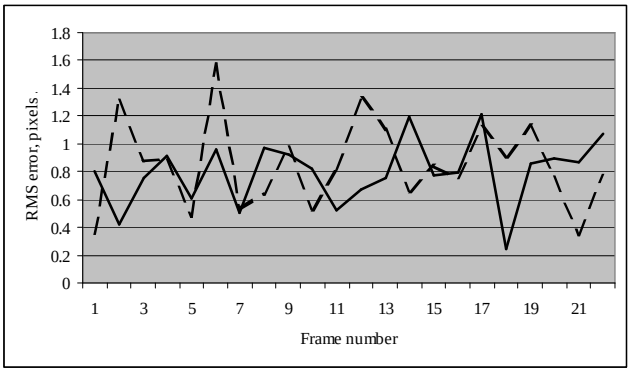

Figure 10 - RMS error for two sets of images used when progressively refining the position estimate

5.2 Initialisation

This section presents results on two aspects of the initialisation processing: the overall performance in terms of the proportion of images from which a successful initialisation can be carried out, and the benefit of retaining spatial information in the Hough transform.

The initialisation process has been used successfully on many hours of both football and rugby coverage, but for the purpose of gathering some statistics on its effectiveness, the first 35 minutes of the test material were studied in detail.

The initialisation process was configured to work with the main camera and the two 18-yard cameras, with the search parameters listed in Table 1. The step size of all search parameters was chosen to be equivalent to a movement of about 10 pixels; in the case of field-of-view, this was the figure at the edge of the image. With this set of parameters, approximately 700,000 different camera poses were searched for each camera position.

camera	*location*	*pan range*	*tilt range*	*field-of-view range*
main	(0.2, 18.5, 75.0)	-65º to 65º	-30º to -5º	4.8º to 40º
left 18-yard	(-34.7 31.7, 90.4)	-80º to 20º	-30º to -5º	4.8º to 40º
right 18-yard	(34.0, 33.2, 93.3)	-20º to 80º	-30º to -5º	4.8º to 40º

Table 1 – Parameters used for initialisation search

An image was grabbed every 5-6 seconds, and the images which showed some of the pitch (250 in all) were included in the test. The results are summarised in Table 2 below.

Result	*Number of occurrences*
Successfully initialised (main camera)	195
Successfully initialised (left 18 yard)	7
Successfully initialised (right 18 yard)	6
Total successfully initialised	**208**
Failed (insufficient lines)	31
Failed (not one of these cameras)	8
Failed (lens angle too tight)	3
Failed (other reasons)	0

Table 2 - Results of initialisation test

From these results, it can be seen that the algorithm initialised successfully for 208 out of the 250 images (83% success rate). The majority of the remaining images contained insufficient lines. Many of these were close-ups on particular players – a kind of shot that it is generally unnecessary to augment with virtual graphics. The next most common cause of failure was images coming from a camera other than the three for which the system had been configured. The remaining cause of failure was very tight zoom values (1.8-3.0º), below the minimum value of 4.8º that had been considered worth including in the initial search.

The total time taken by the initialisation process was 0.99s: 0.02s for locating edge points, 0.33s for computing the spatialised Hough transform (10 spatial subdivisions), and 0.64s for the search (running on a 3.4GHz Pentium 4).

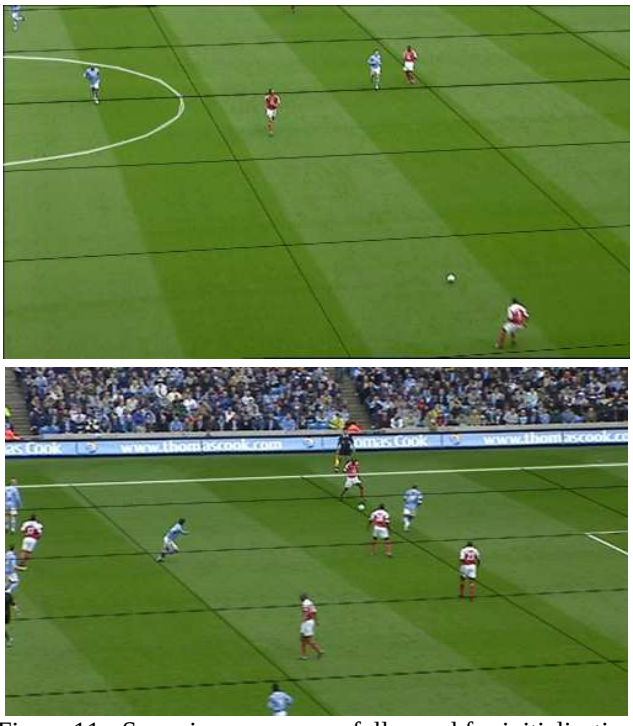

Figure 11 - Some images successfully used for initialisation

Examples of some images containing relatively few lines, that nevertheless the algorithm was able to initialise from, are shown in Figure 11. The lines of the pitch model are overlaid in white; in these images they are very difficult to distinguish from the actual pitch markings, although the line segments used to represent the centre circle are just visible in the upper image. A grid of lines with 10m spacing has been overlaid in black on the ground plane (these should not be confused with the light and dark strips on the pitch itself, which come from the way in which the grass on the pitch was cut, and play no part in the tracking process). The grid overlay is useful when viewing a moving sequence, as it gives an indication of the tracking stability across the whole pitch area.

To illustrate the benefit of retaining spatial information in the Hough transform used during initialisation, the degree to which the correct pose stood out from others was assessed, for several different levels of spatial resolution in the transform.

Figure 12 shows the sum of the bins in the Hough transform corresponding to the expected line positions, for a wide range of pan values, with the position, tilt, roll and field-of-view fixed at values that are approximately correct for the input image shown in Figure 6. The plot shows how the sum varies as the number of spatial subdivisions in the Hough transform is increased. The peak at around –31º corresponds to the true pan angle. With only one subdivision (i.e. a conventional Hough transform), there are several other significant peaks, and the true peak is only 16% higher than the highest of these. When two subdivisions are used, the true peak is 35% higher than the next-highest peak, and this rises to 100% when 10 subdivisions are used. This indicates the usefulness of subdividing each Hough accumulator in order to retain information on the spatial location of contributing samples.

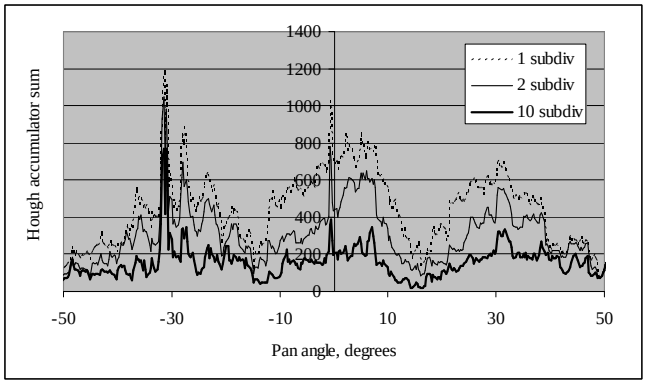

Figure 12 - Sum of Hough accumulator bins as a function of pan angle

The Hough transform of the image of Figure 6 that was used to generate the results of Figure 12 is shown in the left part (a) of Figure 13. The other two transforms in Figure 13 are from the spatial subdivisions corresponding to line segments in the upper or left part of the image (b), and the lower or right part (c) (corresponding to the two-subdivision case plotted in Figure 12). The transform in (a) is the sum of the

transforms (b) and (c). The correspondences between some of the pitch lines and peaks in the Hough transform are shown. Note, for example, that the peaks for lines 2 and 4 have contributions from both spatial subdivisions (as these are long lines covering more than half the image), whereas the peak for line 3 is only visible in the bottom/right spatial subdivision. Conversely, the peak for line 1 is only visible in the top/left subdivision (b), and is easier to spot in this subdivision than in the whole transform (a), as there are less "stray" contributions from other pixels. This illustrates that by examining only the set of spatial subdivisions expected to contribute to a line, a more noise-free measurement is obtained.

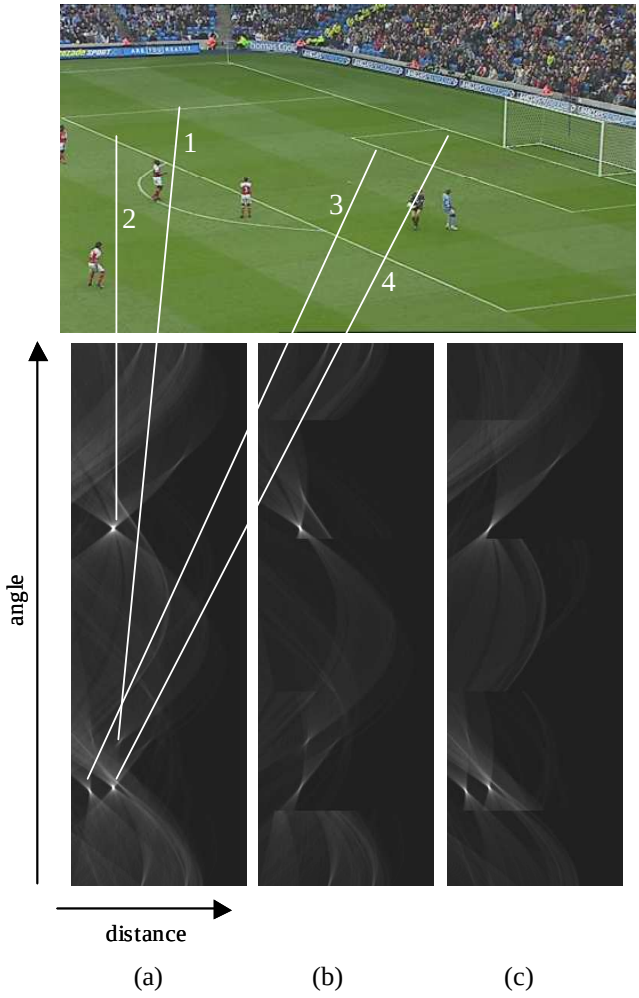

Figure 13 – (a): The complete Hough transform of the pitch lines in Figure 6; (b): just those spatial bins for the top or left half; (c): just those for the bottom or right half.

5.3 Tracking

An example of the pan and tilt values computed for a 20-second part of the test material is shown in Figure 14. This portion of the sequence showed the main camera panning from the right-hand goal towards the centre of the pitch and back again. Without ground truth data it is difficult to demonstrate the accuracy of the results, but it can be seen from the plot that the values vary smoothly, showing that there is very little random noise present. Note that no filtering or smoothing has been applied.

To obtain a rough measure of the noise present in the camera parameters, the acceleration of the pan angle has been plotted in Figure 15. This would be expected to vary smoothly and to remain close to zero, given the smooth way in which cameras are usually panned. It generally lies within 0.02º of zero, suggesting a noise level of about this amount. The vertical field-of-view was around 11º in this sequence, so this level of pan noise would give rise to movement of the overlaid graphics through a distance of about 1 picture line in a 576-line image. The larger peaks in the acceleration are caused by the centre line coming into or out of view, causing a slight shift in computed pose, due to a small difference between the assumed and actual pitch dimensions.

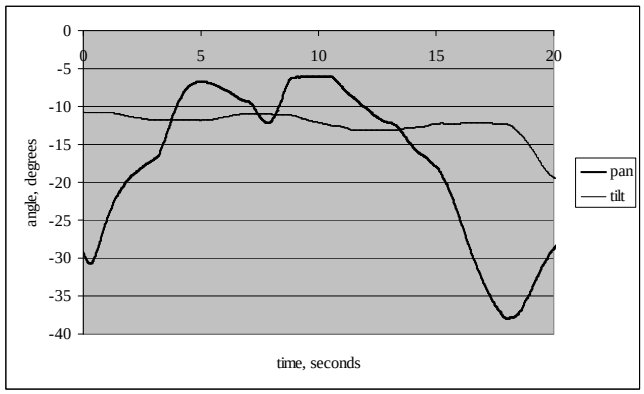

Figure 14 - Pan and tilt angles for a 20s sequence

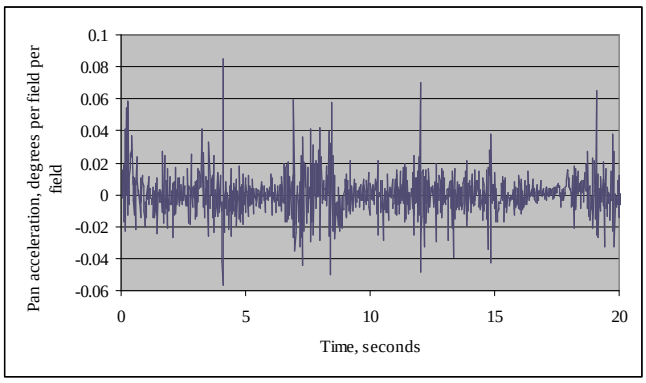

Figure 15 - Second derivative of pan angle

The total processing time for the tracking stage on a typical image was around 5.8ms using a 3.4GHz Pentium 4 processor, although this varied by around ±2ms depending on the number of lines visible. If all detected line points were used in the least-squares optimisation, rather than using the end-points of lines fitted through the points (as discussed in Section 4.2, step 3), the typical processing time rose to around 10.9ms.

5.4 Use with a sports graphics system

The method described here has been incorporated in a commercially-available sports graphics system [8], in order to allow the system to be used without sensors on the cameras. An example of a graphic overlay generated by this system, using the camera tracking data derived using the method described here, was shown in Figure 1.

6. Discussion

The method described here has been found to work well for graphics overlay applications, witnessed by its successful use in a commercial product. It has also been used to calibrate cameras for use in reconstructing 3D models of a football game [4]. For the latter application, calibration accuracy and compensation for lens distortion were particularly important, so the approach mentioned in Section 4.2, in which all detected line pixels are directly used in the minimisation process, was adopted.

One factor that limits the calibration accuracy is the accuracy to which the overall pitch dimensions are known; as mentioned in Section 1, these vary between grounds. The method presented here could be extended to solve for the pitch width and length, by combining information from multiple images in a manner similar to that described in Section 2, although an independent measurement made with a device such as a laser rangefinder would probably give the most accurate answer.

Another aspect of the accuracy of the pitch model concerns the planarity of the pitch. We have so far assumed that the pitch is flat, but in many cases the pitch slopes down towards the edges in order to improve drainage. This could be accounted for either by surveying the pitch to determine its true shape, or by further extending the method presented here to solve for the height variation.

In some situations we have found it useful to include the goal posts in the pitch model, particularly for close-up shots around the goal where relatively few lines may be visible. The goal posts can be difficult to detect reliably, due to the presence of the net and the fact that the background behind them may contain the crowd or advertising hoardings rather than just grass. A more sophisticated line detection method may give improved results.

To improve the robustness and accuracy of tracking when the camera view moves into areas with very few lines, it would be useful to track other image features such as grass texture. This would provide additional information to help changes to pan, tilt and zoom to be estimated, whilst still relying on the appearance of lines at known positions to eliminate long-term drift and provide an absolute scale. An approach based on SLAM (Simultaneous Localisation and Mapping) such as that described in [2] could be used. However, as there is no need to compute the true 3D locations of the additional image features used if the camera position remains fixed, simpler methods of using 2D-3D correspondences are probably more suitable.

7 Conclusion

This paper has presented an algorithm for the real-time computation of the pose of a camera viewing a scene such as a football match. It has been successfully used in a commercial sports graphics system [8], as well as for research into multi-camera 3D reconstruction [4]. Some suggestions for further work to improve the accuracy and robustness have been given.

Acknowledgements

This work was carried out as a part of the EU-funded IST project MATRIS [7]. The author would like to thank his colleagues at BBC Research, and the team at Red Bee Media Ltd, for helpful discussions and the provision of test data.

References

[1] F. Aldershoff, T. Gevers. "Visual Tracking and Localisation of Billboards in Streamed Soccer Matches". Storage and Retrieval Methods and Applications for Multimedia, Proc. SPIE. Vol. 5307, No. 1, pp.408-16, 2004.

[2] A. J. Davison. "Real-time simultaneous localisation and mapping with a single camera". Proceedings of the 9th International Conference on Computer Vision, Nice, 2003.

[3] The Football Association. http://www.thefa.com/TheFA/RulesAndRegulations/FIFALawsOfTheGame/Postings/2002/05/12112.htm

[4] O. Grau, M. Prior-Jones, G.A. Thomas. "3D modelling and rendering of studio and sport scenes for TV applications." WIAMIS 2005, Montreux, Switzerland, April 13-15 2005.

[5] J. A. Heather, X. D. Yang. "Spatial Decomposition of the Hough Transform". Computer and Robot Vision 2005, pp. 476-482.

[6] P.V.C. Hough, "Method and Means of Recognizing Complex Patterns", US Patent 3,069,654, 1962.

[7] The MATRIS project.www.ist-matris.org

[8] Red Bee Media Ltd. The Piero Sports Graphics system. www.redbeemedia.co.uk/piero, www.bbc.co.uk/rd/projects/virtual/piero/

[9] L. Vacchetti, V. Lepetit, P. Fua. "Combining Edge and Texture Information for Real-Time Accurate 3D Camera Tracking". International Symposium on Mixed and Augmented Reality, pp. 48-56, 2004.

[10] T. Watanabe, M. Haseyama, H. Kitajima. "A soccer field tracking method with wire frame model from TV images". 2004 International Conference on Image Processing, pp. 1633-1636.

[11] X. Yu, H.W. Leong, C. Xu, Q. Tian. "A Robust Hough-Based Algorithm for Partial Ellipse Detection in Broadcast Soccer Video". 2004 IEEE International Conference on Multimedia and Expo, pp. 1555-1558.