Automatic Camera Selection for Activity Monitoring in a Multi-camera System for Tennis

Philip Kelly, Ciarán Ó Conaire, Chanyul Kim and Noel E. O'Connor
CLARITY: Centre for Sensor Web Technologies, Dublin City University, Dublin 9, IRELAND
Email: {kellyp, oconaire}@eeng.dcu.ie

***Abstract*—In professional tennis training matches, the coach needs to be able to view play from the most appropriate angle in order to monitor players' activities. In this paper, we describe and evaluate a system for automatic camera selection from a network of synchronised cameras within a tennis sporting arena. This work combines synchronised video streams from multiple cameras into a single summary video suitable for critical review by both tennis players and coaches. Using an overhead camera view, our system automatically determines the 2D tennis-court calibration resulting in a mapping that relates a player's position in the overhead camera to their position and size in another camera view in the network. This allows the system to determine the appearance of a player in each of the other cameras and thereby choose the best view for each player via a novel technique. The video summaries are evaluated in end-user studies and shown to provide an efficient means of multi-stream visualisation for tennis player activity monitoring.**

I. Introduction

In order to be become a successful tennis player one needs to possess a highly proficient and varied skill-set that covers such diverse areas as physical fitness, agility, mental toughness, tactical awareness, and technical ability. It has been shown that the amount of strength, speed, agility and flexibility conditioning a player is prepared to undertake is linked to the standard they play at [1]. However, in younger players, technical stroke production appears to influence rankings more than physical ability [2], and as such their training should concentrate on effective and efficient stroke mechanics, improving technique and ball placement, with less emphasis on physical conditioning. Other work has emphasised that a tactical approach to training (emphasising the role of strategy, tactics and decision making) for younger players leads to better game performance [3].

To be able to teach effectively, a tennis coach must have in-depth understanding of the sport from the fundamental skills to advanced tactics and strategy. In addition, he/she must be able to reveal performance and tactical issues through efficient post-training or post-game analysis and relay this information effectively to the athlete. Video analysis can be an extremely useful medium within the area of sports coaching. This technology provides a means for a coach to effectively identify areas of a player's technique that requires improvement to maximise their technical advancement, and to minimise injury potential. In addition, the medium provides the facilities through which a coach can highlight tactical awareness components of a student's game to increase their cognitive understanding of game playing.

In collaboration with Tennis Ireland [4], the national governing body for the sport of tennis in Ireland, we have developed the TennisSense system [5] at their coaching headquarters. This is a technology platform which aims to provide their coaches with technological solutions that allow them to more effectively develop the next generation of elite tennis athletes. This system provides, amongst other technologies, a number of time-synchronised video cameras located strategically around a tennis court to capture a tennis match from a variety of coach-defined angles.

In this work, we outline a real-time technique for the automatic and intelligent selection of the most appropriate video streams from a camera network and their aggregation into a single coherent video (which temporally combines the most appropriate camera stream for each tennis player on the court). The proposed aggregation methodology can be seen as a two stage process, each of these two stages can be seen as a major contribution to this work. Given a distinct PTZ camera setup, the first stage of this work involves automatically learning the relationships between cameras in the network via a one-time auto-calibration technique. These relationships are then employed in the second contribution of this work, which temporally determines the most descriptive camera viewpoints for each player in a game and accumulates them into a video stream. The resultant video can be automatically generated and can provide a coach with a dedicated review of each player's actions on the court (for both technical or strategical evaluation of performance respectively), which can then be simply and effectively imparted to the student.

The remainder of this paper is organised as follows: Section II outlines previous work in the area of sports video summarisation. We give an overview of the TennisSense platform and provide a high-level overview of the three video aggregation techniques we evaluate in section III. In section IV, we describe the video analysis components that underpin the player tracking techniques and auto-calibration approach of this work. Section V provides quantitative experimental evaluation of the three video aggregation techniques plus a baseline summary technique via a number of user-studies. Finally, we give our conclusions and directions for future work in section VI.

II. Related work

There is an extensive body of research literature concerning the automatic summarisation of broadcast sports matches [6]. The goal is to detect the important parts or *highlights* of the

game, and compile a shorter video clip using only these parts. Many approaches exploit the structure of broadcast video, such as the use of *replays* or crowd-shots, and use these features to determine important events [7], [8]. In this paper, however, we examine the problem of appropriate camera selection from multiple input streams, where the goal is not to reduce the length of the video, but to select a subset of video streams from a larger set of networked cameras. The most straightforward way to do this is selecting the single best camera view at any given time and temporally switching viewpoint as appropriate.

In [9], Wojciechowski et al. propose a general framework for multi-camera stream editing, selecting one stream from multiple input video streams depending on the requirements of the application. Park and Cho [10] examine the problem of summarising event sequences collected in the office environment based on this perspective and propose a fuzzy rule-based approach. They combine camera selection with video summarisation, which reduces the number of streams and also temporal size. However, the annotation of events is done manually, which is time-consuming and impractical for many applications. Lee et al. [11] explore camera selection in a surveillance context and propose a fast sub-optimal algorithm for selecting a subset of cameras that give the best view of the detected faces in the scene. Since their cameras are fully calibrated, they can make inferences as to the overlap in camera views.

In prior work relating to multi-camera calibration, usually a known object is placed at various positions in such a way as to be seen by multiple cameras [12]. Corresponding feature points are extracted to obtain the relative camera positions and orientations. Svoboda et al. [13] propose an automatic method for multi-camera calibration using a video of a moving laser pointer, since this point-light source provides a unique reference point to match in the camera views. In this paper, our approach does not require specific calibration shapes or sequences, but can be obtained by learning correspondences in real video sequences of tennis matches and can overcome the problem of ambiguous matches.

III. Overview

In collaboration with Tennis Ireland [4], we are striving to provide technological solutions to real problems encountered by tennis coaches. As part of this project, we have instrumented one of their indoor tennis courts with data-gathering infrastructure for use as a test-bed for sports and health research. This infrastructure, which is in frequent use by both amateur and elite players, includes nine time-synchronised IP cameras positioned around the court, with pan, tilt and zoom (PTZ) capability (see Figure 1 for the location of the cameras around the court and a sample of the different camera views available) – an overview of the TennisSense infrastructure is detailed in [14]. During specification of the requirements of the system, the coaches requested the installation of an overhead camera with a wide-field of view. While this viewpoint provides little information for the analysis of technical

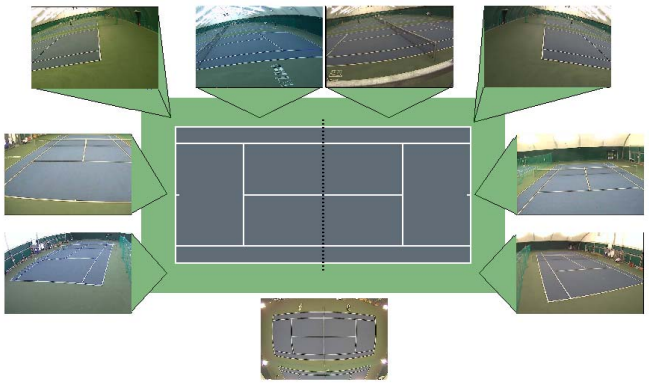


Fig. 1. Camera locations around the court.

ability, it does provide an appropriate viewpoint from which tactical shots and movement during matches can be visualised. Conversely, other camera feeds can provide good viewpoints for technical assessment, but these viewpoints provide much less information about decision making and player strategy. In this work, we outline a technique for the automatic and intelligent aggregation of these multiple camera streams into a single coherent video stream, which provides the viewer with the most appropriate viewpoints that maximises the informational content for each player in the game.

The proposed approach is a two stage process; (1) a one-time auto-calibration technique for a given PTZ camera setup; and (2) the real-time selection and aggregation of a subset of the input video streams, one for each player on the court. In order to select the most appropriate camera feeds at a specific moment in time, we need to be able to assess how a player will appear in each camera view at this point. This process is facilitated by the calibration process (as described in section IV-B). This technique utilises the overhead camera as a common reference to all other networked cameras that are positioned around the tennis court. Using this reference, a mapping is obtained from the (x, y) location of a player in the overhead camera to their size and position on the image plane in every other camera in the framework. This relationship between cameras is learnt automatically and only has to be run once for a given camera setup.

The second stage of the process employs the use of a real-time overhead player tracker (as described in [14], but modified to use our rapid motion detection approach outlined in section IV-A). Using the player tracker, the position and temporal path of each tennis player on the court is determined. Using both the positional and the calibration information, the appearance (size and position) of each player on the court can be evaluated at each time instant. Utilising this knowledge, the best camera stream to display each player on the court can be determined. This process is described in more detail in section IV-C.

IV. Algorithmic Design

In this section, we describe the operation of our camera selection system. Firstly, we explain the motion module we

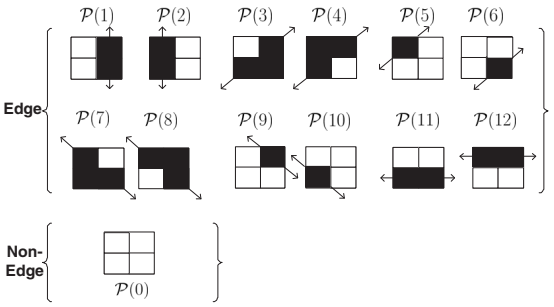


Fig. 2. Edge classification patterns in a 2×2 pixel block; patterns are denoted as numbers according to edge direction (horizontal, vertical, diagonal and negative diagonal)

use to determine the location of the dominant moving object in a camera's field of view. Secondly, we describe how this information can be used to automatically learn a 2D tennis court-calibration which maps player size and position in overhead view to each of the other views. We conclude this section by detailing the steps involved in our proposed approach to multi-stream summary generation.

A. Motion Module

The first step in our motion module is to determine the motion in the image. We use a rapid motion detection procedure developed in [15]. This technique is faster than real-time (shown capable of detecting motion in 500 frames of 384×288 resolution in 1.1 seconds [15]) and therefore can be utilised as an important module within our system for the real-time processing of data. The motion detection consists of two components, edge detection and motion edge detection. An edge is detected by classifying coefficients of the Hadamard Transform (HT) to avoid the computational burden of using conventional edge detection algorithms. The motion edges are extracted by frame differencing the edge map and calculating and tracking the transitions of edge patterns in the same 2×2 block.

1) Edge detection: Edge patterns are assigned identifying numbers by classifying different patterns of pixels in a 2×2 block as shown in Figure 2. We use a sequency ordered Hadamard Transform (HT) to classify edges. Let a radix-N point one-dimensional sequence ordered HT of signal $x(n)$, $n = 0, 1 \ldots N-1$, where N is power of 2, be defined as in Eq 1, which can be easily extended to the two-dimensional case due to its unitary, symmetric property.

$$X(k) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x(n) \prod_{i=0}^{N-1} (-1)^{p_i(n) \times d_i(k)} \tag{1}$$

where $p_i(n)$ is binary representative form of n, $d_i(k)$ can be defined followed by taking binary to gray-code (Eq 3) and bit reversal conversion (Eq 2) of binary representation of k as follows

$$\begin{aligned} n &= (p_{n-1}p_{n-2}\ldots p_0)_2 = \sum_{i=0}^{n-1} p_i 2^i \\ k &= (b_{n-1}b_{n-2}\ldots b_0)_2 = \sum_{i=0}^{n-1} b_i 2^i \end{aligned} \tag{2}$$

$$\begin{aligned} d_0 &= b_{n-1} \\ d_1 &= b_{n-1} + b_{n-2} \\ \vdots &= \vdots \\ d_{n-1} &= b_1 + b_0 \end{aligned} \tag{3}$$

Let the coefficients of the Hadamard transform in a 2×2 image block be denoted as $F(u, v)$. We first calculate the following:

$$\begin{aligned} \chi &= \max(|F(1,0)|, |F(0,1)|, |F(1,1)|) \\ \mathcal{D} &= \begin{cases} \left[\frac{F(1,0)}{F(0,1)} + 0.5\right] & \text{if } F(0,1) \neq 0 \\ 2 & \text{otherwise} \end{cases} \end{aligned} \tag{4}$$

where [] is a round function.

$$\mathcal{P}(i), i = \begin{cases} 0 & \text{if } (\chi \leq \tau) \\ 1 & \text{if } (\chi > \tau \bigcap \mathcal{D} = 2 \bigcap F(1,0) > 0) \\ 2 & \text{if } (\chi > \tau \bigcap \mathcal{D} = 2 \bigcap F(1,0) < 0) \\ 3 & \text{if } (\chi > \tau \bigcap \mathcal{D} = 1 \bigcap F(1,1) > 0 \bigcap F(1,0) > 0) \\ 4 & \text{if } (\chi > \tau \bigcap \mathcal{D} = 1 \bigcap F(1,1) < 0 \bigcap F(1,0) < 0) \\ 5 & \text{if } (\chi > \tau \bigcap \mathcal{D} = 1 \bigcap F(1,1) < 0 \bigcap F(1,0) < 0) \\ 6 & \text{if } (\chi > \tau \bigcap \mathcal{D} = 1 \bigcap F(1,1) < 0 \bigcap F(1,0) > 0) \\ 7 & \text{if } (\chi > \tau \bigcap \mathcal{D} = -1 \bigcap F(1,1) < 0 \bigcap F(1,0) < 0) \\ 8 & \text{if } (\chi > \tau \bigcap \mathcal{D} = -1 \bigcap F(1,1) > 0 \bigcap F(1,0) > 0) \\ 9 & \text{if } (\chi > \tau \bigcap \mathcal{D} = -1 \bigcap F(1,1) > 0 \bigcap F(1,0) > 0) \\ 10 & \text{if } (\chi > \tau \bigcap \mathcal{D} = -1 \bigcap F(1,1) > 0 \bigcap F(1,0) < 0) \\ 11 & \text{if } (\chi > \tau \bigcap \mathcal{D} = 0 \bigcap F(0,1) > 0) \\ 12 & \text{if } (\chi > \tau \bigcap \mathcal{D} = 0 \bigcap F(0,1) < 0) \end{cases} \tag{5}$$

For a non-edge region (assigned $\mathcal{P}(0)$ as shown in Figure 2), the non zero position coefficients $(F(1,0), F(0,1), F(1,1))$ should be zero. Otherwise, the non-zero position coefficients increase according to edge strength. Therefore if χ is greater than a pre-defined threshold value (τ), the block is classified as an edge block, otherwise as a non-edge block. This means that when the difference of pixel values is greater than τ, we consider this block as an edge. Pattern $\mathcal{P}(i)$ is obtained via Eq (4) and the properties of the Hadamard transform, as shown in Eq (5), where τ is a pre-defined threshold value and $\bigcap$ is a logical AND operation.

2) Motion edge detection: After edge detection, the blocks corresponding to motion edges are determined. As an object moves in the scene it covers and uncovers background around its borders and possibly deforms. These phenomena result in a change of the edge characteristics within blocks on the object's

boundary. This can be used to detect motion edge blocks from frame to frame. There are 3 possibilities to consider: (1) edge to non-edge (2) non-edge to edge (3) a change in edge direction.

Within a frame differencing framework, the first possibility above will result in ghost edges that should be removed if they can be detected. In the ideal case, the other possibilities above will result in real motion edges but in practice there will be a lot of noise. To reduce the effect of noise, the history of edges are examined and processed. We introduce the Pixel Bit Mask Difference ($\mathcal{PBMD}$) as an observation factor. $\mathcal{PBMD}$ is the number of different bits between assigned edge patterns. It is increased based on the strength of the noise that potentially results in a bit change. $\mathcal{PBMD}$ can be calculated as follows (where black and white pixels in Figure 2 are set to 0 and 1 respectively):

$$\mathcal{PBMD} = \sum_{(u,v)\in\{0,1\}} \mathcal{P}(i)^{(u,v)} \oplus \mathcal{P}(j)^{(u,v)} \tag{6}$$

where $\oplus$ is a XOR bit operation, $\mathcal{P}(i)^{(u,v)}, \mathcal{P}(j)^{(u,v)}$ are mask bits of $\mathcal{P}(i)$ and $\mathcal{P}(j)$. For example, $\mathcal{PBMD}$ is calculated as '1' (meaning there is one bit difference) between $\mathcal{P}(1)$ and $\mathcal{P}(3), \mathcal{P}(6), \mathcal{P}(8), \mathcal{P}(9)$ by comparing the pixel bit masks. The $\mathcal{PBMD}$ between $\mathcal{P}(1)$ and $\mathcal{P}(0)$ (non-edge block) should be controlled in a different way by setting a maximum value of $\mathcal{PBMD}$.

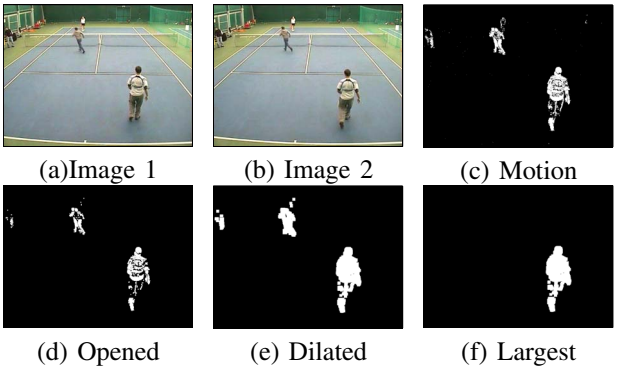
The differencing between edges in temporal sequences sometimes generates false alarms if edges vary in the background. This is an inevitable consequence of frame differencing methods. We introduce a compensation method for false alarms using the History Update Value ($\mathcal{HUV}$) based on $\mathcal{PBMD}$. $\mathcal{HUV}$ is observed at each frame and compared to the $\mathcal{PBMD}$ to classify moving edges from all possible candidate edges. When the current block has the same edge pattern as the previous frame, the $\mathcal{HUV}$ is zero. This means that any edge pattern appearing in this block except the same edge pattern from the previous frame is considered as a moving edge. On the contrary, higher values of $\mathcal{HUV}$ indicate the presence of high levels of noise in past frames, therefore only restricted edge patterns are decided as a moving edge. $\mathcal{HUV}$ can be obtained by using Eq (7). A 2×2 block that has a motion edge is selected by comparing $\mathcal{HUV}$ in the previous frame and the current $\mathcal{PBMD}$.

$$\mathcal{HUV}(i) = \beta \times \mathcal{HUV}(i-1) + (1-\beta) \times \mathcal{PBMD}(i)$$
$$\begin{cases} \mathcal{PBMD}(i+1) > \mathcal{HUV}(i) & \text{moving block} \\ \mathcal{PBMD}(i+1) \leq \mathcal{HUV}(i) & \text{non moving block} \end{cases} \tag{7}$$

where $\mathcal{PBMD}(i), \mathcal{HUV}(i)$ are the $\mathcal{PBMD}$ and $\mathcal{HUV}$ values of a block in the i^{th} frame and β is a weighted constant satisfying the condition $\beta \in [0, 1]$.

In summary, the overall moving edge detection algorithm can be explained as follows.

1) Obtain average pixel differences between frames t and $t-1$ ($ZS^t = (F(0,0)^t - F(0,0)^{t-1})$). If ZS^t is larger than a pre-defined value ($\tau^\star$), a block is considered a candidate moving block, otherwise as a non-moving block.
2) If $ZS^t > \tau^\star$, $\mathcal{PBMD}$ and $\mathcal{HUV}$ are calculated. If $\mathcal{PBMD}(t) > \mathcal{HUV}(t-1)$, this block is considered as a moving block, the value of the $\mathcal{HUV}$ is updated with Eq (7).
3) When moving edge blocks are obtained, moving edge pixels are marked as the whole 2×2 block. This has the effect of generating a sub-sampled moving edge binary image.



(a)Image 1 (b) Image 2 (c) Motion

(d) Opened (e) Dilated (f) Largest

Fig. 3. Motion Module: Our low-complexity motion module computes two-frame motion from (a) and (b), obtaining (c). It then performs a morphological opening to remove noise (d), a dilation with a large structuring element (e) and selection of the largest connected component (f)

Dominant Object Detection After detecting motion using the rapid motion detection procedure, we apply standard image processing techniques to obtain the result. Figure 3 gives an overview of the steps in detection of the dominant object in the camera's view. Given two consecutive images (see Figure 3(a) and (b)), a motion image is generated using the above procedure. To remove noise, we use a morphological opening, followed by a dilation with a 9×9 structuring element to close gaps in the motion mask. Finally we select the largest connected component and store it's bounding box.

B. Automatic Camera Calibration

In this section, we describe the calibration process for the TennisSense camera network streams. For ease of reference, let C_9 represent the overhead camera in the TennisSense framework and let $C_1, C_2 \ldots C_8$ represent the 8 other cameras in the system. Within the calibration stage, each camera is treated independently. As such, we will focus on the process of calibrating one of $C_1, C_2 \ldots C_8$, which we will refer to as C_i. Within the calibration stage, two $2D$ transformations are obtained. When combined, these two mappings correlate the (x, y) pixel position of a tracked tennis player in C_9 and their corresponding appearance (i.e. pixel position and size) in C_i.

The first $2D$ mapping in the calibration process, a $2D$ homography transformation, H, correlates the *groundplane* of the tennis court in C_9 to the corresponding *groundplane* in C_i. As such, the effect of H is to map the (x, y) groundplane pixel position of a tennis player in the overhead camera viewpoint to

their corresponding (x_i, y_i) groundplane pixel position in C_i. In this work, the groundplane pixel position is taken to mean the position on the ground directly below a tennis player's centre of gravity – i.e. the midpoint of the bottom of the bounding box around the player in question. Further details regarding to the calculation of H will be presented in the next section of this paper.

The second $2D$ mapping in the calibration process takes as an input the (x_i, y_i) groundplane pixel position in C_i and determines the height of the player in pixels in C_i's imageplane. This process is similar to the one presented in [16], where a linear model of an average person's $2D$ image height to their $2D$ image foot location (both in pixels) is obtained. This mapping can be created automatically using the dominant object regions obtained in a calibration sequence. In this approach, the foot positions are defined, as before, as the midpoint of the bottom of the bounding box around the regions. Similarly, the head position are defined as the midpoint of the *top* of the bounding box. Using these foot and head positions, a line can be automatically fit to the data using a least squares error technique. It should be noted, that this approach makes the assumption that the horizon appears approximately horizontal in C_i.

Given these two $2D$ mappings, we are able to model the position and appearance of a player's bounding box in camera C_i's imageplane if we know their (x, y) position in C_9's imageplane. This calibration process is far simpler than full $3D$ camera calibration techniques, however the relationships obtained are still powerful enough to meet our application the requirements. In addition, the proposed approach can be learnt automatically, which is advantageous as the tennis coaches may change the PTZ camera parameters between or during games, and consequently a complex manual calibration process is best avoided.

In the proposed approach, we track the required (x, y) positions of tennis players in C_9 via a modified tracker of that detailed in the work of [14]. In this work, the tracker is modified by the use our rapid motion detection module (as outlined in section IV-A) instead of background subtraction. This alteration reduces the computational complexity of the tracking module and increases robustness to changes in lighting which can occur on the court of play. For the remainder of this section we will focus in more detail on how we generate the first $2D$ mapping in the calibration process, namely H.

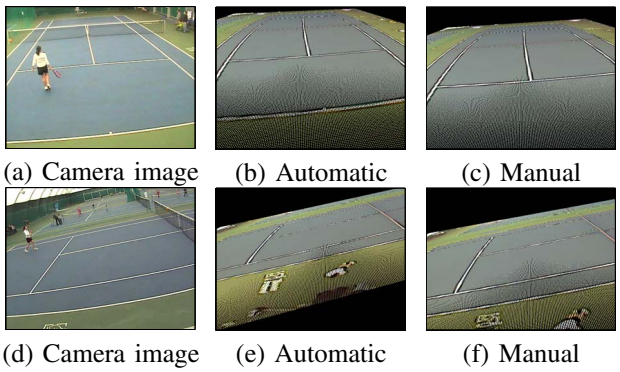
Homography Generation

In the first stage of this process, potential correspondences between tracked players in C_9 and motion regions detected in C_i's imageplane are obtained from a calibration sequence. To detect motion regions, the process for dominant object detection is employed. The calibration process takes place over a number of frames, during which one or more players are tracked in C_9 and can be seen (at least for some of the frames) in C_i. This sequence can be simply obtained from a tennis match being played out in front of the cameras in the TennisSense network, as such no calibration shapes or predefined motions are required for this process. For each frame in the calibration sequence, a list of potential correspondences are added to a table. This table consists of potential associations between each of the tracked player positions in C_9 (of which there may be many) and the bounding box of the dominant object detected in C_i for the corresponding temporally aligned frame (for which there will only be one per frame).

The dominant object is likely to have been caused by *only one* of the tracked players in C_9. As such, if more than one player is tracked in C_9, many of the potential correspondences in the table are likely to be outliers. In addition, it was found that since a player spends most of their time on or near the court baseline, the correspondences were unevenly distributed over the tennis court space. This uneven distribution could cause a bias in the estimation of H. To combat these issues of significant outliers and player positional bias, the correspondences are pre-processed to determine a *single* correspondence between dominant motion regions and each (x, y) location in the overhead camera view. Specifically, for each 8×8 region in C_9, a single corresponding rectangle is chosen in C_i to represent the entire set of correspondences found within this area.

For the 8×8 region, all potential correspondences are combined using RANSAC [17]. One score function we investigated was to maximise the number of inliers, where an inlier is defined to be that which has over 80% overlap with the bounding box. However, using a simple inlier count did not prove to be an adequate choice, since only dominant object regions are utilised – as such, if a given player, P_1, is in view and tracked in C_9 but there is a second player, P_2, closer to C_i than P_1, then P_1 will only appear as a dominant region when P_2 is out of C_i's field of view. As such, in the table of correspondences, the tracked position of P_1 in C_9 will more often be incorrectly associated with the dominant region of P_2.

In order to overcome the issue of having a high number of such outliers, two weighting priors are associated to each bounding box. The RANSAC approach then seeks to maximise the sum of the inliers' weights. The first prior adds weight to rarely occurring dominant regions of activity, the second includes a weighting for region size. The first prior is learnt from *all* motion regions detected in the camera view by simply counting the number of times a pixel occurs within a motion region. Then, to compute a prior for a motion region, we determine the average count of pixels in the region and compute the inverse as the prior weighting. The second weighting prior is calculated via $1/(1 - P(s > s_{obj}))$, where s_{obj} is the bounding box area. This probability distribution is similarly learnt from all motion regions detected in the current view. Using RANSAC the correspondence which maximises the sum of inliers' weights is selected. The dominant region of this correspondence is combined with all dominant regions in the inlier set by simple averaging of their bounding box coordinates. The result of this averaging is then used to represent the *single* correspondence between the dominant motion region in C_i and the centroid of the 8×8 region in



(a) Camera image (b) Automatic (c) Manual

(d) Camera image (e) Automatic (f) Manual

Fig. 4. Court calibration example images: (a) and (d) are Camera images, (b) and (e) show estimated ground-planes using our automatic approach, (c) and (f) show a ground-plane model created by manually selecting points.

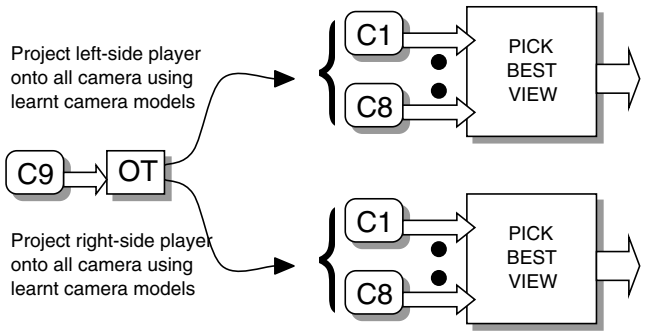


Fig. 5. Flowchat of proposed video aggregation technique (tracking 2 players in a singles match): C_i represents camera stream i, with C_9 as the overhead camera. OT is the overhead (person) tracker.

the overhead view.

Finally, we run RANSAC on the reduced set of correspondences to compute H from the tracked (x, y) position of a player in C_9 image coordinates to a set of correlated image positions (x_i, y_i) in C_i. Some results of this automatic calibration of $2D$ homography transformations can be seen in Figure 4. In Figures 4 (b), (c), (e) and (f) the resultant images are obtained by taking an image obtained from C_9 and transforming it via H. As such the closer Figures 4 (b and c)/(e and f) are to (a)/(d), the better the calibration of H.

C. Summary Generation

Figure 5 illustrates our proposed summary generation approach. Unlike in other sports, such as soccer, where a single camera view can be used to observe the ball in play, choosing the best view to follow the tennis ball movements would be disconcerting for the viewer as the camera view would change too rapidly. Our goal is to obtain the best camera view for each player and to create a two-view summary, as illustrated in Figure 6(b). We track tennis players using the overhead camera using the tracker as outlined in Section IV-B and remove any tracked people that are not within the play area (defined as the court-area plus a border around the baselines). We then select one person from each side of the court, giving priority to people with longer track histories to avoid tracking ball boys. If no one is detected in the left or right play areas, then the overhead view (camera C_9) is selected as the default view. Otherwise, we use the learnt models (from Section IV-B) to project each player into each of the other 8 camera views, and compute a bounding box indicating where they will appear in the scene. We compute a weighting, W, to give a larger score to cameras that have the player in the centre of their field of view. We used $W = exp(-(D/\sigma)^2)$, where D is the distance from the bounding box centroid to the centre of the image and we set σ so that $W \approx 0.5$ at the image border. A camera's score is simply calculated as the bounding box area multiplied by the weighting. We accumulated each camera's score over 25 frames (1 second) and the camera with the highest score was selected for viewing the player over that 25 frame time period. We investigated other metrics for assessing view quality, such as estimating the player's direction of motion and assuming that a fast movement towards a camera indicated that the tennis stroke would be seen from this camera, but that proved not to be the case. By simply using the estimated area of the player's body as a metric, we ensure the majority of the player is seen at the highest possible resolution.

V. Experimental Results

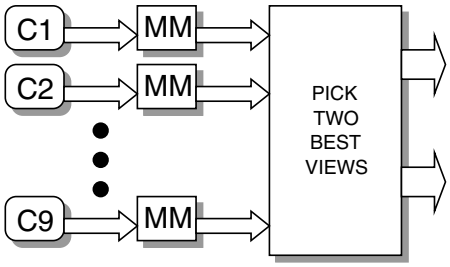
To evaluate our video aggregation approach, we performed a number of user-studies. We applied our algorithm to three 1-minute clips of test video, each sequence consisting of two players in a competitive match setting – see Figure 6(b) for an example of frames from the resultant video. In addition, three further video sequences for each test video sequence were generated as follows; (1) a *baseline* technique was applied, whereby the video streams from $C_1, C_2 \ldots C_8$ were simply tiled into a single video – see Figure 6(a); (2) an approach (see figure 7(a)), which we will refer to as *Method A*, whereby all cameras were treated independently and the two highest scoring camera views were chosen to be incorporated into the final video; and (3) a technique (see figure 7(b)), which we will refer to as *Method B*, whereby$C_1, C_2 \ldots C_8$ were manually clustered into two distinct groups of four, based on what side of the net the camera was mainly focused on, and the highest scoring camera view in each cluster was chosen to be incorporated into the final video. For both *Method A* and *Method B*, the score for each camera was obtained by firstly obtaining the dominant motion region in each frame, then multiplying the bounding box area of this motion region by the same weighting as defined in section IV-C. Note that this is different to the proposed approach, which does not detect the motion in each camera, but instead predicts the motion by projecting the detected people from the overhead view. For both these techniques, if no movement was detected in any frame, the camera viewpoint defaulted to the overhead camera viewpoint (as was the heuristic of our proposed approach). In all test sequences, each output sequence was set to have a resolution and frame-rate of 1280×480 and 25Hz respectively. A total of 12 users evaluated all summaries.

The user studies were implemented as follows. Three sets of four 1-minute clips of test videos, each set consisted of a

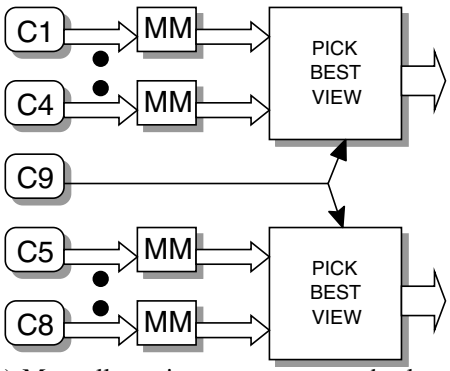
(a) Tiling of 8 camera views (baseline approach)

(b) Selection of the two best views

Fig. 6. Examples of frames from summary videos clips shown to users: (a) Simple camera tiling from baseline video, (b) Two camera selection method, used in all other summaries.



(a) Pick the best 2 views from all cameras



(b) Manually assign cameras to each players

Fig. 7. Diagrams of summary generation methods that we compared to our proposed approach: C_i represents camera stream i, with C_9 as the overhead camera. MM is the motion module from section IV-A and OT is the overhead (person) tracker.

baseline technique, which was always played first to the user in order for them to get a full overview of the events in the clips, and three other techniques, namely *Method A*, *Method B* and our proposed approach (which we will refer to as *Method C*). These latter three approaches were shown in a random order to prevent bias. After viewing each video, the user was asked to rate for the sequence video clip within three areas.

1) **Ease of observing activities** – How well were you able to observe all important activities in the video?
2) **Correct selection of camera view** – Do you feel the video allowed you to view the appropriate camera angle for understanding the activities in the video?
3) **Comfortable viewing experience** – How comfortable did you find the experience of viewing the video?

For each question, the user was asked to rate the video from 1 to 7, where 1 was the best score possible. In addition, the user was allowed to enter additional comments about each sequence if desired.

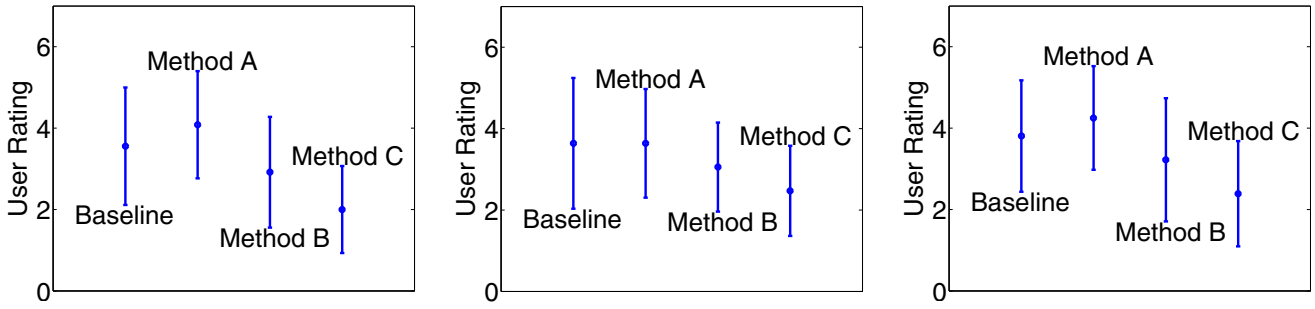
A. Discussion of results

The results of our user study are shown in figure 8. In the three metrics, the proposed approach (Method C) performs best, followed by Method B. Method A appears slightly worse than the baseline.

The comments we received from users were quite helpful in assessing the relative merits of the different approaches. One criticism of the proposed approach was that sometimes the tennis ball or the player's racket was not seen (*'sometimes the ball disappears from the screen (too high), and I would feel more comfortable seeing the ball all the time'*). Since we estimate only the position of the player's body, we do not have any motion information on the racket or ball movement. On the other hand, our method only needs to process the overhead video stream, instead of processing the motion in all cameras. In future work, we intend to do a full 3D calibration of the cameras in order to track the ball in 3D, which will allow improved accuracy of view selection.

Method B performed well, but the main criticism was that there was *'too much movement between camera angles/views'*. Since it is based on motion, background movement can affect its performance if the tennis players are not generating much motion.

Users liked to be able to see both players on screen and to have fixed sides for the players (i.e. one player on the left and one on the right). By choosing the two largest motion regions, summaries generated by Method A appeared confusing to users, as it would often show the same player on both selected views, or would switch sides of the screen for a player. On the other hand, both Methods B and C had this functionality. Most users would follow play by tracking the ball and looking at either the left or right view: *'I find myself alternating from watching L then R then L then R as the play progresses, and that is a comfortable model for me to follow - I actually feel I am in control, more so than when watching a single screen playback of a tennis match, as on TV'*.



(a) Ease of observing activities (b) Correct selection of camera view (c) Comfortable viewing experience

Fig. 8. User study results: lower scores indicate a higher user satisfaction (1 = Very Good, 7 = Very Bad). For each set of results, the dot represents the mean and the line represents the standard deviation. For example, in (a) the Baseline method had a mean of 3 and a standard deviation of approximately ± 1.

The baseline method, where we showed all 8 camera simultaneously (see Figure 6(a)), was found to provide too much information to the user and users found it confusing; *'Too many camera views to see in one video at the same time'*, *'so many videos displayed concurrently made it difficult to know where to look'* and *'2 screens is my max comfortable zone'*. Despite this, one user found the baseline to be less confusing with repeated viewing of this type of summary: *'Having used this view for the third time, I found I got used to it and didn't find it as confusing as before'*. Since the baseline avoids switching views, the trade-off is a lower resolution of each view and many users commented on this: *'The scenes are now too small to comfortably follow play and the other 7 views are distracting and tough on the eyes.'*.

VI. Conclusions and Future Work

In this paper we have presented a method for automatic camera selection from multiple video streams in a sporting scenario. Using an overhead view, our system can automatically learn the $2D$ relationships between each camera view and an overhead camera view. This model allows us to project a player tracked from overhead into the other camera views and predict how well their activities would be seen. We compared our camera selection approach to several alternative approaches in user studies. Users found that summaries generated using our proposed approach were more comfortable to watch, and allowed them to better monitor the activities during play.

The summarisation approach proposed in this paper does not temporally shrink the length of the video. As such, the removal of redundant parts of the video, such as removing parts of the video where the ball is not in play [14], is targeted as future work. Additionally, in this work we have assumed that both players are equally important, but it could be that a particular player is the focus of the training and the goal is therefore to select cameras that give the best view of that player (e.g. front and back view). The calibration and overhead detection that are already part of our system allow for this possibility and we intend to investigate this also.

Acknowledgment

This work is supported by Science Foundation Ireland under grant 07/CE/I1147.

References

[1] P. Roetert, S. Brown, P. Piorkowskil, and R. Woods, "Fitness comparisons among three different levels of elite tennis players," *Journal of Strength and Conditioning Research*, vol. 10, no. 3, pp. 139–143, 1996.

[2] M. Kovacs, "Applied physiology of tennis performance," *British Journal of Sports Medicine*, vol. 40, pp. 381–386, 2006.

[3] A. Turner, "A comparative analisis of two approaches for teaching tennis: Game based approach versus technique approach," in *2nd ITF Tennis science and technology Congress*, 2003.

[4] "Tennis ireland," http://www.tennisireland.ie/.

[5] "Tennissense," http://www.cdvp.dcu.ie/tennisireland/.

[6] D. Sadlier and N. O'Connor., "Event detection in field sports video using audio-visual features and a support vector machine," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 10, pp. 1225–1233, 2005.

[7] Z. Zhao, S. Jiang, Q. Huang, and G. Zhu, "Highlight summarization in sports video based on replay detection," in *ICME*, 2006, pp. 1613–1616.

[8] D. Zhong, R. Kumar, and S.-F. Chang, "Real-time personalized sports video filtering and summarization," in *MULTIMEDIA '01: Proceedings of the ninth ACM international conference on Multimedia*. New York, NY, USA: ACM, 2001, pp. 623–625.

[9] K. Wojciechowski, B. Smolka, H. Palus, R. Kozera, W. Skarbek, and L. Noakes, "Multi camera automatic video editing," *Computer Vision and Graphics*, vol. 32, no. 10, pp. 935–945, 2006.

[10] H.-S. Park and S.-B. Cho, "A fuzzy rule-based system with ontology for summarization of multi-camera event sequences," in *ICAISC*, 2008, pp. 850–860.

[11] H. Lee, L. Tessens, M. Morbee, H. Aghajan, and W. Philips, "Sub-optimal camera selection in practical vision networks through shape approximation," in *International Conference on Advanced Concepts for Intelligent Vision Systems*, 2008, pp. 266–277.

[12] W.-C. F. John Kassebaum, Nirupama Bulusu, "Smart camera network localization using a 3d target," in *ImageSense: Workshop on Applications, Systems, and Algorithms for Image Sensing*, Nov 2008.

[13] T. Svoboda, D. Martinec, and T. Pajdla, "A convenient multi-camera self-calibration for virtual environments," *PRESENCE: Teleoperators and Virtual Environments*, vol. 14, no. 4, pp. 407–422, August 2005.

[14] C. Ó Conaire, P. Kelly, D. Connaghan, and N. E. O'Connor, "Tennissense: A platform for extracting semantic information from multi-camera tennis data," in *International Conference on Digital Signal Processing (DSP)*, 2009.

[15] C. Kim and N. O'Connor., "Using the discrete hadamard transform to detect moving objects in surveillance video," in *VISAPP 2009 - International Conference on Computer Vision Theory and Applications*, 2009.

[16] P. Kelly, C. Ó Conaire, and N. E. O'Connor, "Exploiting contextual data for event retrieval in surveillance video," in *ACM International Conference on Image and Video Retrieval (CIVR)*, July 2009, pp. 8–10.

[17] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.