A new model based approach for tennis court tracking in real time

Manel Farhat*, Ali Khalfallah and Med Salim Bouhlel

Research Unit: Sciences and Technologies of Image and Telecommunications,
Higher Institute of Biotechnology,
University of Sfax,
Sfax,Tunisia
Email: manel.farhat@setit.rnu.tn
Email: ali.khalfallah@enetcom.rnu.tn
Email: medsalim.bouhlel@enis.rnu.tn
*Corresponding author

Abstract: The detection and the tracking of the tennis court is a primordial step to analyse a tennis video at higher semantic level. In this context, a new approach for tennis court tracking in real time is proposed in this paper. Our proposed system is based on model based approach allows to compute the homography between the court detected in the scene and the court model presenting the real world coordinate. For this aim, the first step is to detect the tennis court by detecting the court line and determining some interest points. We check then the motion of the camera. In case of camera motion, the court is tracked by tracking the interest points using the Lucas-Kanade algorithm. After that, these points are used by a RANSAC algorithm to estimate the homography. However, in case of a fixed camera, we need only the model based correction system.

Keywords: tracking; RANSAC; model fitting; homography; Lucas-Kanade tracker.

Reference to this paper should be made as follows: Farhat, M., Khalfallah, A. and Bouhlel, M.S. (2018) 'A new model based approach for tennis court tracking in real time', *Int. J. Signal and Imaging Systems Engineering*, Vol. 11, No. 1, pp.9–19.

Biographical notes: Manel Farhat received his Engineering Diploma in Electronics in 2011 and the Master in Electrical Engineering in 2013 from the University of Sfax. Currently, she is a PhD student. Her research interests include image processing and computer vision.

Ali Khalfallah received his Diploma of Engineer in Electricity in 2003 and the Diploma of Master in 2004 from the University of Sfax. He obtained the Doctorate in Electronics in 2008. Currently, he is an Assistant Professor at the Higher Institute of Electronics and Communications of Sfax-Tunisia. His research interests include digital watermarking, cryptography, image fusion, image expansion and image processing.

Med Salim Bouhlel received his Engineering Diploma from the University of Sfax in 1981; the DEA in Automatic and Informatics in 1981 and the degree of Doctor in 1983 from the University of Lyon. He is actually a Professor at the Higher Institute of Biotechnology of Sfax. His research interests include signal, image and video processing, image compression, tomography, and human-computer interaction.

1 Introduction

A huge amount of sports video databases are generated every day. This rapid growth requires efficient and effective tools to provide an automatic annotation which receive, recently, much attention. Owing to advances in scientific researches in different fields such as signal processing, machine learning (Hadj Mabrouk, 2016; Loussaief and Abdelkrim, 2016, and computer vision, building different tools and automatic annotation techniques has become possible (Kijak et al., 2003; Kolonias et al., 2004; Huang et al., 2009 and Yan et al., 2014).

The sports video analysis generates many potential applications such as enhanced broadcast (Owens et al., 2003 and Han et al., 2007), event detection and classification (Kapela et al., 2015), summarisation (Mendi et al., 2013) and virtual advertisement...(Chang et al., 2010).

Sports videos which have the largest share of scientific researches are court games such as soccer and tennis, not only because they are the most popular, but also because they have well-structured rules that facilitate their analysis.

An Automatic content analysis of tennis videos at a higher semantic level requires a robust and efficient

detection and tracking algorithm. However, the existing state of the art approaches and methods of object tracking (Jinan and Raveendran, 2016; Elafi et al., 2016; Collins et al., 2005) cannot be useful in sports video due to their highly dynamic nature. Thus, a successful sport automatic content analysis system must exploit specific information of the game (white court-lines, uniform court colour..) rather than use a generic solution.

In court games, to analyse the video, it is required to determine the ball or players positions in the real-world coordinates of the court. For this purpose, it is very important to detect and track the court to estimate the relation between the image coordinates and the real-world coordinates. As the playfield is planar, this relation can be presented by a projective transformation. Hence, the transformation parameters are estimated by finding a set of correspondences between the tracked court in the scene video and a court model presented in the real-world coordinates.

In early work, Sudhir et al. (1998) have described a court-line detection algorithm based on straight-line detection method in order to build up a tennis court model. The method proposed needs four predefined points on a tennis court to calibrate the camera and estimate the projective transformation. Thus, the main disadvantage of this algorithm is that it must be manually initialised and it is also not robust against the occlusions of these four predefined points. It does not guarantee good results if we have a partial occlusion of the court. One more robust detection of soccer court is presented in Kim and Hong (2001) and Watanabe et al. (2004), however, it used an exhaustive search for the parameter space which is computationally complex. In Ekin et al. (2003) authors proposed a Hough transform method which allows detecting shots showing the goal in soccer videos without any estimation of the projective transformation between the real-world coordinates and the image coordinates.

Dang et al. (2010) used a RANSAC-based line detection algorithm to detect the court line. They made the assumption that the speed of the camera change is small and they extended the detected court in the previous frame to define a search region. Then, to track the model, the same detection method applied in the new local search area is used. Chang et al. (2010) applied a standard Hough transform to extract court line and define the intersection point. However, it chooses only four line intersections to estimate the perspective transformation. In Lai et al. (2011), authors were based on Harris Corner Detector (Mikolajczyk and Schmid, 2004) to extract the court feature and the scale-invariant feature transform (SIFT) (Lowe, 2004) to present the descriptor of each detected point.

In this context, This paper presents a new automatically court tracking system for tennis video allowing to estimate the projective transformation between the court in the image coordinate and a court model in the real world coordinate. The major contribution in this paper is that it presents an automatically algorithm for tracking tennis court by combining Lukas-Knades tracker system (LK) (Tomasi and Kanade, 1991; Lucas and Kanade, 1981), the Random Sample Consensus algorithm (RANSAC) (Fischler and Bolles, 1981), the Direct Linear Transformation algorithm (DLT) (Hartley and Zisserman, 2004) and finally the Levenberg-Marquardt (LM) minimisation algorithm (Press and Flannery, 1988) that fits the model projected to the court detected in the image. In addition, the proposed approach takes into consideration the movement of the camera which improves the stability of the court detection.

2 Proposed method

The proposed system for court detection and tracking is illustrated in Figure 1. Our objective is to track a court from a tennis video and to estimate the homography which is necessary to convert the image coordinates of the video frame to the real-world coordinates. For this aim, we firstly detect the frames containing the court by applying a court view detection process. Once the current frame is a court view frame, we proceed to the initialisation step. It starts with segmenting the scene and extracting all pixels of the court lines in order to detect straight lines in the image. Then, we found a set of interest points and some putative correspondences. Thus, we used RANSAC algorithm to estimate the geometric transformation between the image coordinate and the model. This transformation is applied to project the court model into the image. For each geometric transformation, we estimated the error projection between the white court line in the image and the projected model. A transformation with a low error projection is selected as an optimal solution.

Once the court is detected for the first frame, we tested the motion of the camera. In case of camera motion, we will be based on Lukas-Knades tracker in order to track the interest points and we will use Ransac to estimate the homography for each frame. Finally, to ensure a high performance, a model based correction process is applied to reduce the projection error using the Levenberg-Marquardt (LM) minimisation algorithm.

However, in case of a fixed camera, we need only to apply the model based correction process.

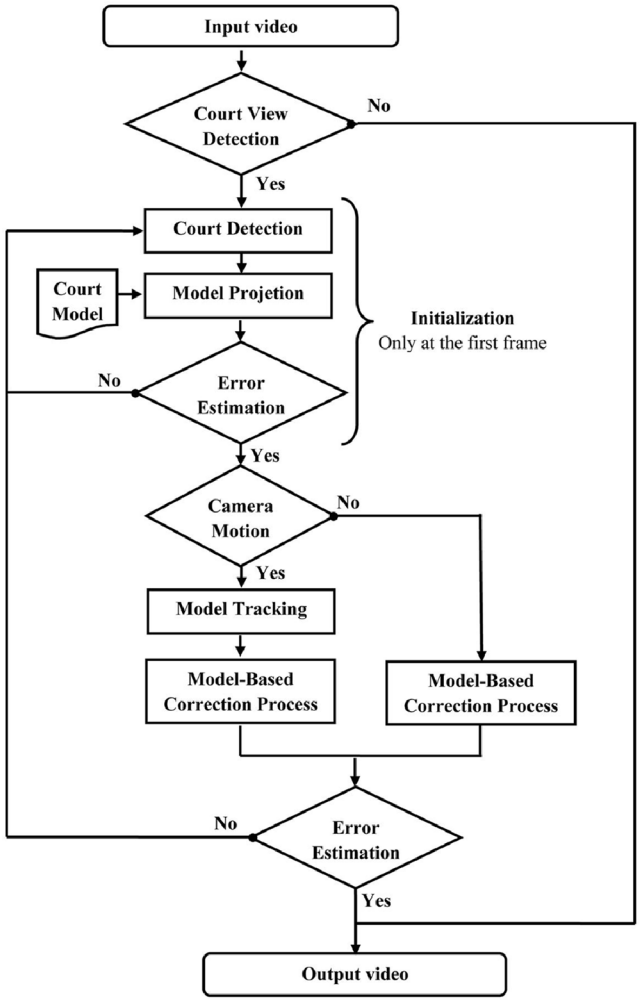
2.1 Court view detection

While observing a tennis video, we can notice that the camera typically switches to wide shots presenting the court view just before the serve, and it keeps this position throughout the shot, and between the shots camera focus on other different plans such as spectator or players...etc. So, detecting the court view is a primordial step before detecting and tracking the court.

In the tennis video, a court view has usually a unique hue histogram distribution. So, based on this observation and to recognise the court view, we used one court view hue-histogram as a pattern to query the tennis video looking for the court view frame with applying a histogram similarity measures. Then, we can classify the current frame to court view or non-court view frame by being based on the

histogram similarity measures. If it is less than an empirically fixed threshold, the frame will be considered as a court view shot and we can proceed with the court detection step. Hence the remaining step of our proposed algorithm will be applied only if the current frame is a court view frame (Farhat et al., 2017).

Figure 1 Illustration of our proposed approach



2.2 Camera motion detection

For the camera motion detection, we used a method inspired by Bradski and Davis (2000). The first step is to generate the silhouette of the moving objects in the scene. Many classical methods of silhouette generation can be used such as stereo depth subtraction (Beymer and Konolige, 1999), colour histogram back-projection (Bradski, 1998), infra-red back-lighting (Davis and Bobick, 1998) and background subtraction (Martins et al., 1999; Elgammal et al., 2000) that needs a background model. In our case, we chose the simple method frame differencing (Davis and Bobick, 1997).

Based on the sequentially fading silhouettes, we can record the history of all previous movements which are referred to as motion history image (MHI). MHI representation is a scalar-valued image, where the intensity is directly related to the recent of the motion: the pixels having the most recent movement are the brighter. Examples of MHIs presentation are shown in Figure 2(a) and 2(c).

The final element of the camera motion detection is the temporal segmentation. We segment the MHI presentation in order to group motion regions produced by the movement of the scene objects. Thus, to do this, the MHI presentation is firstly scanned for the silhouette regions with the most recent timestamp. Once the region having the most current timestamp is detected, its perimeter is searched. Then, a flood fill algorithm is applied to isolate the region of the motion found.

In case of fixed camera (Figure 2(a)) only the motion of the player is detected. However, in case, where the camera is moving, we can notice the global motion of the whole image (Figure 2(c)).

2.3 Court detection

2.3.1 Image segmentation

Since the colour of the court lines is always white, we chose to segment the entire scene to keep only the white pixels. For this, we applied the method of simple thresholding to the image in greyscale (Lafi et al., 2016). After the segmentation step, we obtained a binary image (Amri et al., 2017) as shown in Figure 3(a). It can be observed that the result contains other white pixels than these of the court lines such as white logo, player or spectators wearing in white and so on. These non-court line pixels affect the accuracy of the line detection step. Thus, a post-processing step is essential. To remove rebel pixels, firstly, we filtered our binary image making some criterion for blobs feature, we removed structures having a large area and structures having a short contour. Then, in order to improve results and extract vertical and horizontal line pixels, we used morphological operations with a corresponding structure element. We applied opening (erosion followed by dilation) to the binary image with structure element 1×7 all-ones matrix to extract horizontal line pixels (Figure 3(c)) and 5×1 all-ones matrix for vertical line pixels (Figure 3(b)).

2.3.2 Line detection

2.3.2.1 Line parameter estimation

After the segmentation step, the Hough transform (Sere et al., 2012) is performed on the previously obtained binary images in order to detect court lines. With Hough transform, a line is characterised by two parameters (Θ, ρ) where Θ is the angle between normal line leading to the origin and the x-axis and ρ presents the length of the normal line. The Hough transform constructs an accumulator matrix representing parameter space $\theta_{\max}$ columns and $\rho_{\max}$ rows. For each point of the binary image, ρ is calculated for all θ value, varying between 1 and 180, and for each found ρ the accumulator must be incremented by 1. Then, lines are detected by extracted local maxima in the accumulator that

are above a certain threshold. The main advantage of Hough transform is the robustness to outliers, it can be valuable to detect lines having some short breaks due to the noise or occlusion. Thus, lines detected need not necessarily to be continuous. However, the disadvantage of Hough transform is the detection of a bundle of lines for one targeted line. This is due to the thickness of the lines in the binary image or to the bad quality of the video. To address this issue, we introduced a supplementary step that computes the optimum line.

Figure 2 Example of camera motion detection: (a) MHI in case of fixed camera; (b) original image; (c) MHI in case of camera motion and (d) camera motion detected (see online version for colours)

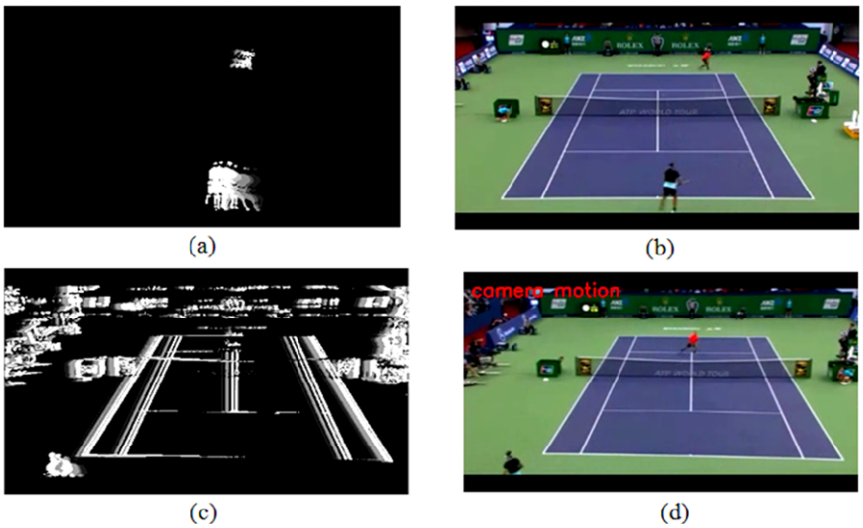
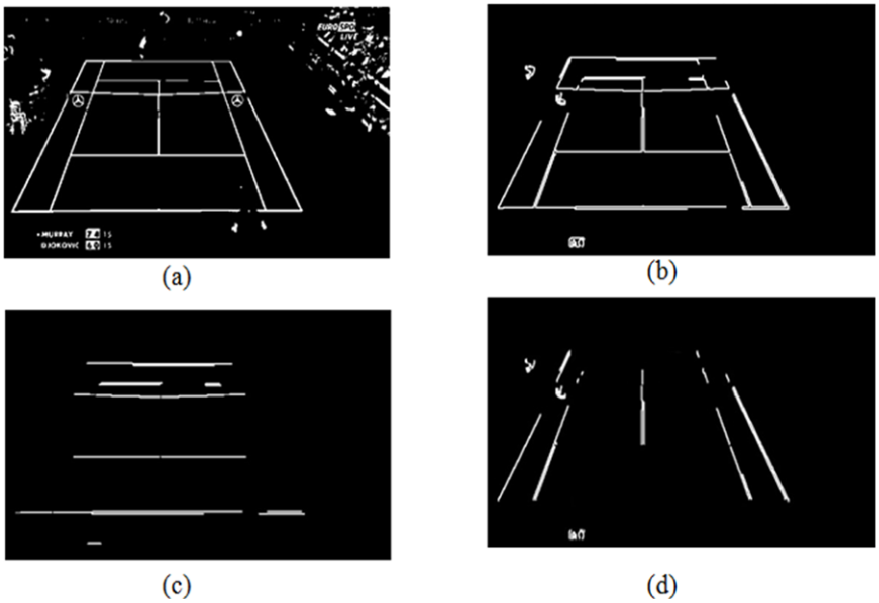


Figure 3 Thresholding of court lines and noise removal and thinning: (a) the segmentation result of the court view; (b) removed rebel pixels result; (c) horizontal lines extraction and (d) vertical lines extraction



2.3.2.2 Line refinement and classification

As we have already mentioned, Hough transform produces some neighbouring lines that belong to one court line. To solve this problem, we removed lines having nearly equal parameters and we kept only one. First of all, we kept only horizontal and vertical lines and we classified them. A line having two endpoints coordinates ($x1$, $y1$) and ($x2$, $y2$) is regarded as a horizontal line if $|y1 - y2| < 10$ and have a pent very near to '0' and it is considered as a vertical line if $|x1 - x2| < 10$. Then, we removed all duplicated lines and kept the median. Two lines L1 and L2 are duplicated if the angle $(\widehat{L1, L2}) < 0.75°$ and the distance $d(L1, L2) < 5$ (Dang et al., 2010).

After we classified the detected lines and removed duplicated ones, we can easily determine the intersection of each pair of horizontal and vertical lines using a basic vector math.

In Figure 4, vertical lines are drawn in red and horizontal lines are drawn in yellow and the intersection points are represented as blue circles.

Figure 4 Thresholding result of the line detection process: (a) lines extraction by Hough transform; (b) result after lines refinement and (c) detection of intersection points (see online version for colours)

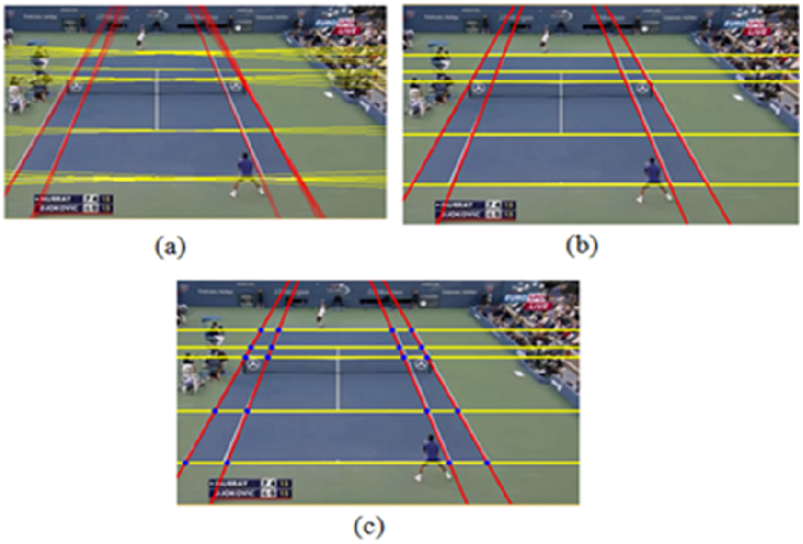
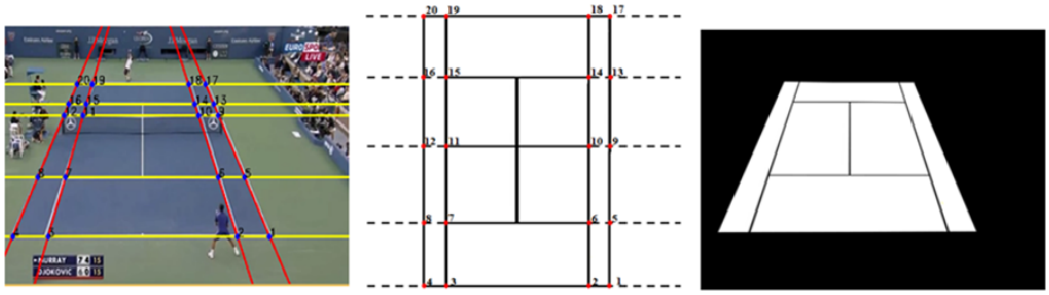


Figure 5 Points correspondences between model and image: (a) interest point in the image; (b) interest points in the model and (c) projected model (see online version for colours)



2.4 Model projection

A court model is a set of lines drawn onto the ground and defining the playfield geometry. These lines are presented in the model coordinate system. The world coordinate system is defined as a Cartesian coordinate system presenting the orientation and the position of the model. So, the two coordinate systems can handily be equal. Thus, model fitting consists of estimating the homography between the real world coordinate and the image coordinate system. For that, we must go through two steps. We need in the first step to find interest points and find putative correspondences, then we estimate the homography by RANSAC algorithm.

2.4.1 Interest point and finding correspondence

As we have already mentioned in Section 2.3.2.2, we detected the court lines and sorted them into two sets vertical and horizontal lines and we also determined the intersection points which will present the interest points of our system. To determine the correspondence of our interest points in the model, we ordered the set of the vertical line from the right to the left and the horizontal lines from the top to the bottom and we numerated the intersection points (Figure 5(a)). Then, to determine the correspondences between the model (Figure 5(b)) and the intersection points we put the constraint where the order is preserved.

2.4.2 Homography estimation

Our detected interest points can include some outliers which prevent a correct estimation of the homography. Thus, we used RANSAC algorithm to identify perfect subsets of correspondences to obtain a better homography.

2.4.2.1 Random sample consensus (RANSAC)

Unlike many of popular robust estimation methods such as least-median squares and M-estimators that are proposed by the statistics literature and then adopted by the community of computer vision, RANSAC was developed by the computer vision community specifically to meet the needs of computer vision applications. RANSAC is a resampling technique to estimate the parameters of a model and fits this data by random sampling of observed data. It uses a small subset of the input data and then it enlarges this set with some consistent data (Fischler and Bolles, 1981).

This data is supposed to be all inliers, and a model is fitted to these provided values and then all other data are fitted to this estimated model. If a sufficient number of data

points is found as inliers, the estimated model is redone using all newly found inliers instead of just using the original subset based on a smoothing method like the least-squares smoothing. The distance between the inliers and the model is calculated, it represents the estimated model's error, and if this error is below a defined threshold, then the estimated model is optimal. However, if it is above the threshold, another subset of the input data points will be selected and the process is restarted.

The idea to estimate homography using RANSAC algorithm is very simple, however, it is powerful. Since we need at least 4 points to estimate the homography, we selected randomly 4 points from the set of all interest points detected and we computed the homography using these points and the Direct Linear Transformation algorithm. Then, we see if this homography agrees with other interest points up to a fixed threshold and we determine the inliers correspondences. This process will be repeated until that we find the best homography with the largest support, and it will be considered as the robust fit.

2.4.2.2 Direct linear transformation (DLT)

The direct linear transform (DLT) (Hartley and Zisserman, 2004) is a method which solves the homography matrix H using a sufficient set of variables. For a point correspondence $X_1 \leftrightarrow X_2$ and a homography H, we have the camera projected image $X_2 = H.X_1$, H is a 3×3 matrix with h_i^T is the ith row of H. Then we have:

$$X_2 = HX_1 = \begin{pmatrix} h_1^T & X_1 \\ h_2^T & X_1 \\ h_3^T & X_1 \end{pmatrix} \text{ where } X_2 = (x_2, y_2, w_2)^T \tag{1}$$

We can express this equation as:

$$X_2 - HX_1 = \begin{pmatrix} y_2 h_3^T X_1 - w_2 h_2^T X_1 \\ w_2 h_1^T X_1 - x_2 h_3^T X_1 \\ x_2 h_2^T X_1 - y_2 h_1^T X_1 \end{pmatrix} = 0. \tag{2}$$

Since $h_i^T.X_1 = X_1^T.h_i$, where $i = \{1,2,3\}$, equation (3) can be written as:

$$\begin{bmatrix} 0^T & -w_2 X_1^T & y_2 X_1^T \\ w_2 X_1^T & 0^T & -x_2 X_1^T \\ -y_2 X_1^T & x_2 X_1^T & 0^T \end{bmatrix} \begin{pmatrix} h_1 \\ h_2 \\ h_3 \end{pmatrix} = 0 \tag{3}$$

We have only two independent equations, So, the third one will be omitted:

$$\begin{bmatrix} 0^T & -w_2 X_1^T & y_2 X_1^T \\ w_2 X_1^T & 0^T & -x_2 X_1^T \end{bmatrix} \begin{pmatrix} h_1 \\ h_2 \\ h_3 \end{pmatrix} = 0 \tag{4}$$

Equation (4) can be written with the form $A_i h = 0$ where h represents a 9×1 vector consisting of the entries of the homography H and A_i is a 2×9 matrix.

If we have $n (n \geq 4)$ correspondences, we obtain a matrix A with $2n \times 9$ dimension and we have in this case an over-determined system. We must then solve the system $Ah = 0$ using a least-squares method. We can write the solution of h as minimise$||Ah||$ with the constraint $||h|| = 1$ The solution of h is, therefore, the unit singular vector associated with the smallest singular value of A ($A = \text{UDV}^T$). It can be easily computed using a singular value decomposition method (SVD) (Strang, 2006; Lafi et al., 2016).

The DLT algorithm is a less well-conditioned problem. So, to assure numerical stability, a normalisation process is required.

The Normalised algorithm steps are as follows:

Normalise the points X_1 with the similarity transformation T_1 (translation and scaling) to denote a new point $\tilde{X}_1 (\tilde{X}_1 = T_1.X_1)$ which has a centroid at the origin with (0, 0)T coordinate, and an average distance from the origin equal to $\sqrt{2}$.

Apply similarly an analogous normalisation process to the correspondence point X_2 using a similarity transformation T_2, we have then the transformed points $\tilde{X}_2 (\tilde{X}_2 = T_2.X_2)$.

Apply the standard DLT algorithm to the normalised point correspondences $\tilde{X}_1$ and $\tilde{X}_2$ to compute the homography $\tilde{H}$.

Denormalise $\tilde{H}$ to get the desired homography H.

$$X_2 = HX_1 = T_2^{-1} \tilde{X}_2 = T_2^{-1} \tilde{H} \tilde{X}_1 = T_2^{-1} \tilde{H} T_1 X_1. \tag{5}$$

Finally, we have:

$$H = T_2^{-1} \tilde{H} T_1. \tag{6}$$

2.5 Error estimation

After the homography estimation, we transformed all points (Pm) of the court model to the image coordinates using the estimated homography H, $P_I(x_i,y_i) = \text{HP}_m$; where P_I represents the pixel of the model projected into the source image.

We are required to verify that the model, projected onto the current frame, covers accurately the court in the image. For this aim, we tested the values pixel in the segmented image for each projected model point P_I. If the projected point is not a court pixel and presents a black pixel, the error value E ($E = (1 - \sum_i E_i)$) increases by 1 if not, E does not change (Equation 7).

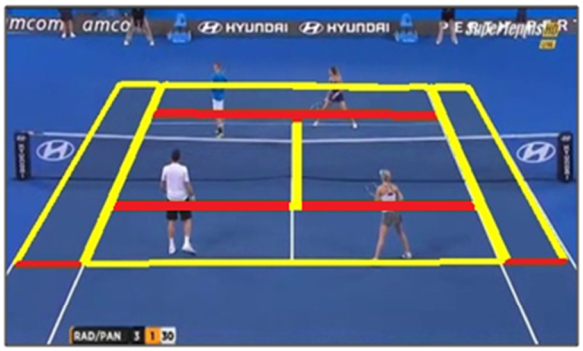
$$E_i = \begin{cases} 1, & \text{if } I(x_i, y_i) = 0 \\ 0, & \text{if } I(x_i, y_i) = 1 \end{cases} \text{ and } \varepsilon = 1 - \frac{\sum_i E_i}{i}, \tag{7}$$

where i is the total number of model pixels and ε is the estimated error.

Each Homography computed is rated by the rejection test; if the estimated error is lower than an experimentally fixed threshold, the homography is selected as an optimal solution and we can move to the next step to track the

model, if it is not the case, the initialisation process must be repeated (Figure 6).

Figure 6 Evaluation of model match (see online version for colours)



2.6 Model tracking

The previous court model initialisation algorithm should be applied only in the bootstrapping process for every new playing shot. Once the interest points are detected, we use the feature-tracking algorithm Lucas-Kanade (LK) (Salehpoor and Behrad, 2012) to track them. The Lucas Kanade algorithm is an often used differential method for optical flow estimation.

Optical flow is generally used to compute the motion of all pixels in a sequence of images (Doyle et al., 2014 and Smith et al., 2010). It presents the apparent motion of the brightness pattern in the images. It works on the assumptions that the pixel intensities are constant between two consecutive frames and the neighbourhood pixels have a similar motion.

Let the intensity of an interest point $I(x, y, t)$, move in the scene and after time dt the point displacement is (dx, dy).

So, if we use Taylor series for $I(x, y, t)$, we can write the following equation:

$$I(x+dx,y+dy,t+dt)=I(x,y,t)+\frac{\partial I}{\partial x}dx+\frac{\partial I}{\partial y}dy+\frac{\partial I}{\partial t}dt+\cdots \quad (8)$$

And, assuming the brightness constancy and according to the assumption that the Brightness of all point is invariable in time, we can write:

$$I(x,y,t)=I(x+dx,y+dy,t+dt) \quad (9)$$

So, equations (8) and (9) give:

$$\frac{\partial I}{\partial x}dx+\frac{\partial I}{\partial y}dy+\frac{\partial I}{\partial t}dt+\cdots=0. \quad (10)$$

Dividing by dt we obtain:

$$-\frac{\partial I}{\partial t}=\frac{\partial I}{\partial x}\frac{\partial x}{\partial t}+\frac{\partial I}{\partial y}\frac{\partial y}{\partial t} \quad (11)$$

where $\frac{\partial x}{\partial t}=V_x$ an $\frac{\partial y}{\partial t}=Vy$ the field components of optical flow $\vec{V}$, respectively, in x and y coordinates. This equation is usually called an optical flow constraint equation. So, calculating the optical flow for each pixel in the image sequence returns to calculate the following equation:

$$-\frac{\partial I}{\partial t}=V_x\cdot\frac{\partial I}{\partial x}+V_x\cdot\frac{\partial I}{\partial y} \quad (12)$$

$$-I_t(p)=I_x(p)\cdot V_x+I_y(p)\cdot V_y \quad (13)$$

Equation (13) has a known variables $I_x(p)$ and $I_y(p)$ which are the image gradients, $I_t(p)$ the gradient along time and two unknown variables (V_x, V_y). There are several methods proposed to solve this problem, one of them is the Lucas-Kanade algorithm. The Lucas-Kanade method assumes that all the neighbouring pixels have similar motions. Tacking a 3x3 windows centred at p, we have so 9 points having the same motion. Now we obtain a system of nine equations with two unknown variables that can be solved using the least squares criterion.

$$\begin{bmatrix}V_x\\V_y\end{bmatrix}=\begin{bmatrix}\sum_i I_x(p_i)^2 & \sum_i I_x(p_i)I_y(p_i)\\ \sum_i I_x(p_i)I_y(p_i) & \sum_i I_y(p_i)^2\end{bmatrix}^{-1}\begin{bmatrix}-\sum_i I_x(p_i)I_t(p_i)\\-\sum_i I_y(p_i)I_t(p_i)\end{bmatrix} \quad (14)$$

The main advantage of Lucas Kanade method is the very fast calculation and the accurate time derivatives. However, as all point tracker algorithm that progresses over time, Lucas Kanade can lose some points due to occlusion, lighting variation, articulated motion or out of the plan. To solve this problem, RANSAC is used for each frame to compute the appropriate homography from the tracked interest points. Then, to ensure a high performance for our system, a model based correction process is applied.

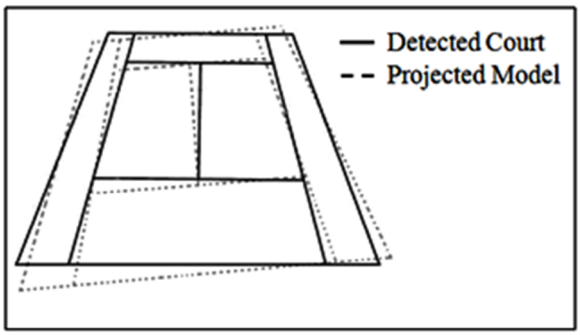
2.7 Model based correction system

The Model based Correction process is applied to the resulting homography estimated by RANSAC algorithm in order to reduce the projection errors. It consists to fit the projected model to our court in the real scene and reduce the projected error. For this aim, a local search is applied to each model point projected to find the closest white neighbourhood pixel in the segmented image which is considered as a court pixel. So, the court model points are grouped with the closest white pixels found in the segmented image. Figure 7 shows how to fit the model to our court detected in the image. We note white pixels in the segmented image with $P_i=(x_i,y_i,1)^T$ and their corresponding in the model with $M_i=(x'_i,y'_i,1)^T$. The intention is to determine the refined homography H' by minimising the Euclidean distance between the model and all the white court pixels in the segmented image. The total symmetric transfer error D is defined as:

$$D=\sum_i(d(P_i,HM_i)^2+d(M_i,H^{-1}P_i)^2). \quad (15)$$

Thus, to minimise D and to find the refined homography we used the Levenberg-Marquardt (LM) minimisation algorithm (Press and Flannery, 1988).

Figure 7 Fitting a projected model to our court in the image



3 Experimental results

In this section, we are interested in evaluating the performances of our proposed system. We have tested our approach to 10 tennis videos with different surface types: hard, clay and grass court (Figure 8). Each video has 512 by 288 resolution, a frame rate equal to 25 fps and a duration about 10 min.

In our evaluation, we were based on a frame-based metrics provided by Bashir and Porikli (Bashir and Porikli, 2006). We used three sets of the tracking evaluation metrics which are, the Tracker Detection Rate (TRDR), the False Alarm Rate (FAR) and the Accuracy.

$$TRDR = \frac{TP}{GT} \tag{16}$$

$$\text{FAR} = \frac{FP}{TP + FP} \tag{17}$$

$$\text{Accuracy} = \frac{TN + TP}{TF} \tag{18}$$

where

- *True positives* (*TP*): represents the number of frames where the court is firmly detected.
- True negatives (*TN*): represents the number of frames where the system approves correctly the absence of the court like the ground truth.
- *False positives* (*FP*): represents the number of frames where the system approves incorrectly the presence the court.
- *Ground truth* (*GT*): represents the ground truth information.
- *Total frame* (*TF*): represents the total number of frames in the tennis video.

Table 1 shows that our system achieved an impressive average rate TRDR and Accuracy of court detection (respectively 0.9733 and 0.9831) with a very low rate FAR 0.0448. This efficient performance of the system is due to track the court only in case of moving camera and also the update of the homography in each frame using the RANSAC algorithm and the combination of Levenberg-Marquardt minimisation algorithm and the error estimation to correct any wrong detection (Figure 9). In addition, the system is very robust to partial occlusion of the court (Figure 10(a)) and some other particularly difficult scenes such as (Figure 10(b)) which contains a shadow in the scene court.

Since claiming that the proposed system working in a real-time, it is important to discuss this time performance. In practice, the speed of a system is relative to the video quality and the computer performance. The timing results presented in this section are calculated for 320 x 240 video resolution and a standard PC (Intel Core i7-4720HQ 3.6GHz).

Table 2 detail the timing for all relevant steps of the proposed algorithm and Figure 11 present the total computational time for all the system.

Figure 8 Visual experimental results in different court types (see online version for colours)

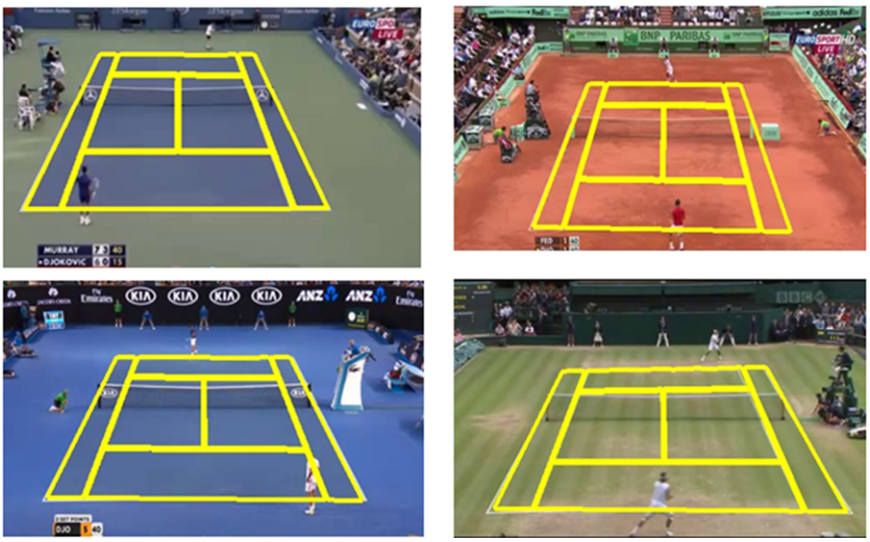


Table 1 Evaluation of court tracking results

Sample	*Tournament*	*Surface*	*TRDR*	*FAR*	*Accuracy*
#1	Australian Open 2016	Hard	0.985	0.013	0.991
#2	ATP World TOUR 2016	Hard bi-colour	0.987	0.0022	0.9886
#3	Hopman Cup 2014	Hard	0.982	0.027	0.9815
#4	ATP World Tour EASTBOURNE	Grass	0.981	0.0148	0.983
#5	Wimbledon	Grass	0.981	0.018	0.989
#6	ATP World Tour Quito 2015	Clay	0.9742	0.021	0.977
#7	Rome 2014	Clay	0.9814	0.0223	0.98
#8	ATP Qatar Open 2016	Hard bi-colour	0.9722	0.0278	0.99
Average rate			0.980	0.018	0.985

Table 2 The average time for all relevant steps

	Court view detection	*Camera motion*	*Court detection*	*Model projection*	*Model tracking*	*Error estimation*	*Model based correction system*
Time (ms)	4.83	4.57	34.69	32.72	23.23	1.04	3.22

Figure 9 Example of error estimation and tracking precision (see online version for colours)

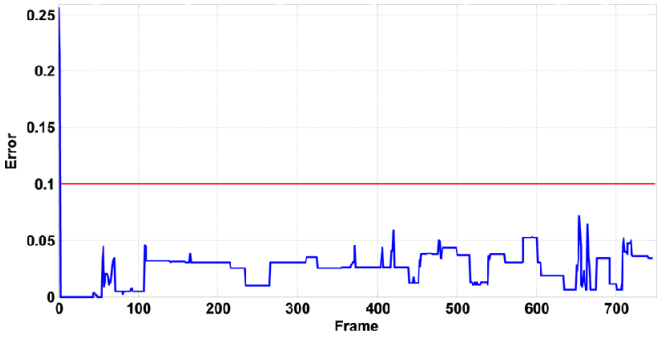


Figure 10 Court detection in a scene with some complication: (a) occlusion and (b) strong shadow (see online version for colours)

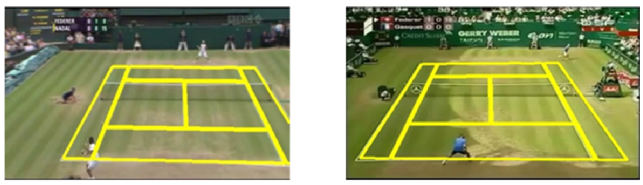
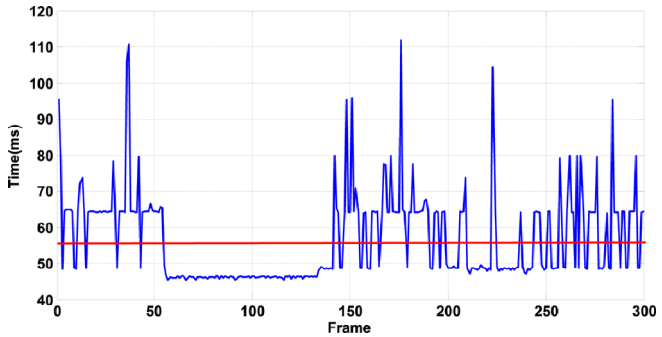


Figure 11 Example of the total computational time for all the system (see online version for colours)



The result shows that our system runs smoothly in real-time. However, the initialisation process can be seen as a little time-consuming process. But this does not harm the time performance of our system because it will be only applied in the bootstrapping process. The total computational time achieved an average of 56.84 ms.

4 Conclusion

In this paper, we have described a new automatic system for tennis court view tracking take into consideration the movement of the camera. The first step in the proposed algorithm is to extract the court view frame based on histogram similarity method. Secondly, we applied, for the court view frame, a court detection based on court line detection to initialisate our system. Once the court is detected for the first frame, we checked the camera movement. If the camera moved, a tracking algorithm using an LK method and a RANSAC homography estimation would be applied to track the movement of the court. Then, we used a model based correction system using Levenberg-Marquardt minimisation algorithm and the error estimation step. However, for the opposite case (fixed camera), the position of the court does not change. So, we need just the model based correction system. Finally, experimental results show that the proposed system is very effective and suitable for any type of tennis court surfaces. To track the court only in case of moving camera provides for the system more stability. Hence, our system achieves robustness and efficiency by combining a set of different methods that are not sufficient on their own.

Acknowledgement

This work was supported and funded by the Ministry of Higher Education and Scientific Research of Tunisia.

References

Amri, H., Khalfallah, A., Lapayre, J-C. and Bouhlel, M.S. (2017) 'Medical image compression approach based on image resizing, digital watermarking and lossless compression', *The Imaging Science Journal*, Vol. 65, No. 2, pp.98–107.

Bashir, F. and Porikli, F. (2006) 'Performance Evaluation of Object Detection and Tracking Systems', *Proc. Ninth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance* (*PETS 2006*) Vol. 5, pp.7–14.

Beymer, D. and Konolige, K. (1999) 'Real-time tracking of multiple people using stereo', *IEEE FRAME-RATE Workshop*, Colchester, UK, pp.1–8.

Bradski, G. (1998) 'Computer vision face tracking for use in a perceptual user interface ', *Intel Technology Journal, Q2'98*, Princeton, NJ, USA, January, pp.214–219.

Bradski, G. and Davis, J. (2000) 'Motion segmentation and pose recognition with motion history gradients', *IEEE Workshop on Applications of Computer Vision*, Palm Springs, CA, USA, 4–6 December, https://doi:org/10.1007/s001380100064

Chang, C., Hsieh, K., Chiang, M. and Wu, J. (2010) 'Virtual spotlighted advertising for tennis videos', *Journal of Visual Communication and Image Representation*, Vol. 21, No. 7, pp.595–612.

Collins, R., Liu, Y. and Leordeanu, M. (2005) 'Online selection of discriminative tracking features', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 27, No. 10, pp.1631–1643.

Dang, B., Tran, A., Dinh, T. and Dinh, T. (2010) 'A real time player tracking system for broadcast tennis video', *Intelligent Information and Database Systems*, Vol. 5991, pp.105–113.

Davis, J. and Bobick, A. (1997) 'The representation and recognition of human movement using temporal templates', *The 1997 Conference on Computer Vision and Pattern Recognition* (*CVPR'97*), Washington DC, USA, 17–19 June, pp.928–934.

Davis, J. and Bobick, A. (1998) 'A robust human-silhouette extraction technique for interactive virtual environments', *Proceedings Modelling and Motion Capture Techniques for Virtual Environments*, Geneva, Switzerland, 26–27 November, pp.12–25, https://doi.org/10.1007/3-540-49384-0_2

Doyle, D., Jennings, A. and Black, J. (2014) 'Optical flow background estimation for real-time pan/tilt camera object tracking', *Measurement*, Vol. 48, pp.195–207.

Ekin, A., Tekalp, A. and Mehrotra, R. (2003) 'Automatic soccer video analysis and summarization', *IEEE Transactions on Image Processing*, Vol. 12, No. 7, pp.796–807.

Elafi, I., Jedra, M. and Zahid, N. (2016) 'Unsupervised detection and tracking of moving objects for video surveillance applications', *Pattern Recognition Letters*, Vol. 84, pp.70–77.

Elgammal, A., Harwood, D. and Davis, L. (2000) 'Non-parametric model for background subtraction', *6th European Conference on Computer Vision*, 26 June–1 July, Dublin, Ireland, pp.751–767.

Farhat, M., Khalfallah, A. and Bouhlel, M.S. (2017) 'A new approach for automatic view detection system in tennis video', *International Journal of Signal and Imaging Systems Engineering*, 2017, Vol. 10, No. 4, pp.195–203.

Fischler, M. and Bolles, R. (1981) 'Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography', *Communications of the ACM*, Vol. 24, No. 6, pp.381–395.

Hadj Mabrouk, H. (2016) 'Machine learning from experience feedback on accidents in transport', *7th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT)*, Hammamet, Tunisia, 18–20 December, 2016,

Han, J., Farin, D. and De With, P.H.N. (2007) 'A Real-Time Augmented Reality System for Sports Broadcast Video Enhancement', *15th ACM International Conference on Multimedia MM*'07, September 24–29, pp.337–340.

Hartley, R. and Zisserman, A. (2004) *Multiple View Geometry in Computer Vision*, 2nd ed., Cambridge University Press, New York, USA.

Huang, Y., Chiou, C. and Sandnes, F. (2009) 'An intelligent strategy for the automatic detection of highlights in tennis video recordings', *Expert Systems with Applications*, Vol. 36, No. 6, pp.9907–9918.

Jinan, R. and Raveendran, T. (2016) 'Particle filters for multiple target tracking', *Procedia Technology*, Vol. 24, pp.980–987.

Kapela, R., Aleksandra, Ś., Rybarczyk, A., Kolanowski, K. and O'Connor, N.E. (2015) 'Real-time event classification in field sport videos', *Signal Processing: Image Communication*, Vol. 35, pp.35–45.

Kijak, E., Gravier, G., Gros, P., Oisel, L. and Bimbot, F. (2003) 'Hmm based structuring of tennis videos using visual and audio cues', *The IEEE International Conference on Multimedia and Expo*, Vol. 3, pp.309–312.

Kim, H. and Hong, K. (2001) 'Robust image mosaicing of soccer videos using self-calibration and line tracking''*Pattern Analysis and Applications*, Vol. 4, pp.9–19.

Kolonias, I., Christmas, W. and Kittler, J. (2004) 'Tracking the evolution of a tennis match using hidden Markov models', *Joint IAPR International Workshops on Structural, Syntactic and Statistical Pattern Recognition*, Mérida, Mexico, 29 November–2 December, pp.1078–1086, https://doi.org/10.1007/978-3-540-27868-9_119

Lafi, S., Khalfallah, A., Bouzid, M., Bouchot, A. and Bouhlel, M.S. (2016) 'Blind source separation based on ICA algorithm applied to multispectral fluorescence imaging', *International Review on Computers and Software*, Vol. 11, No. 3, pp.192–199.

Lafi, S., Khalfallah, A., Bouzid, M., Bouchot, A. and Bouhlel, M.S. (2016) 'A cholesterol lesion detection approach based on SVD decomposition', *7th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications* (*SETIT 2016*), 18–20 December, Tunisia.

Lai, J., Chen, C., Kao, C. and Chien, S. (2011) 'Tennis video 2.0: a new presentation of sports videos with content separation and rendering', *Journal of Visual Communication and Image Representation*, Vol. 22, No. 3, pp.271–283.

Loussaief, S. and Abdelkrim, A. (2016) 'Machine learning framework for image classification', *7th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT)*, Hammamet, Tunisia, 18–20 December, 2016, https://doi:org/10.1109/SETIT.2016.7939841

Lowe, D. (2004) 'Distinctive image features from scale-invariant keypoint', *International Journal of Computer Vision*, Vol. 60, No. 2, pp.91–110.

Lucas, B. and Kanade, T. (1981) 'An iterative image registration technique with an application to stereo vision', *The International Joint Conference on Artificial Intelligence*, Vancouver, BC, Canada, 24–28 August, pp.674–679.

Martins, F., Nickerson, B., Bostrom, V. and Hazra, R. (1999) 'Implementation of a real-time foreground/background segmentation system on the intel architecture', *IEEE International Conference on Computer Vision Frame Rate Workshop*. Kerkyra, Greece, September.

Mendi, E., Clemente, H. and Bayrak, C. (2013) 'Sports video summarization based on motion analysis', *Computers and Electrical Engineering*, Vol. 39, No. 3, pp.790–796.

Mikolajczyk, K. and Schmid, C. (2004) 'Scale and affine invariant interest point detectors', *International Journal of Computer Vision*, Vol. 60, No. 1, pp.63–86.

Owens, N., Harris, C. and Stennett, C. (2003) 'Hawk-eye tennis system', *Proc. Int'l Conf. Visual Information Eng. (VIE'03)*, Guildford, UK, 7–9 July, pp.182–185, https://doi.org/10.1049/cp:20030517

Press, W., Flannery, B., Teukolsky, S. and Vetterling, W. (1988) *Numerical Recipes in C*, Cambridge University Press, New York, USA, https://doi:org/10.1017/CBO9780511811685.

Salehpoor, M. and Behrad, A. (2012) '3D face reconstruction by KLT feature extraction and model consistency match refining and growing', *6th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT)*, Sousse, Tunisia, 21–24 March, 2012, https://doi:org/10.1109/SETIT.2012.6481932

Sere, A., Andres, E. and Sie, O. (2012) 'Extended standard Hough transform for analytical line recognition', *6th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT),* Sousse, Tunisia, 21–24 March, 2012, https://doi:org/10.1109/SETIT.2012.6481950

Smith, M., Boxerbaum, A. and Peterson, G. and Quinn, R. (2010) 'Electronic image stabilization using optical flow with inertial fusion', *The IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, pp.1146–1153, https://doi:org/10.1109/IROS.2010.5651113

Strang, G. (2006) *Linear Algebra and its Applications*, 4th ed., Thomson, Brooks/Cole, Cengage.

Sudhir, G., Lee, J. and Jain, A. (1998) 'Automatic classification of tennis video for high-level content-based retrieval', *International Workshop on Content-Based Access of Image and Video Databases (CAIVD'98)*, Bombay, India, 3 January, 1998, pp.81–90, https://doi:org/10.1109/CAIVD.1998.646036

Tomasi, C. and Kanade, T. (1991) *Detection and Tracking of Point Features*, Carnegie Mellon University Technical Report CMU-CS-91–132.

Watanabe, T., Haseyama, M. and Kitajima, H. (2004) 'A soccer field tracking method with wire frame model from TV images', *IEEE International Conference on Image Processing* (*ICIP'04*), 24–27 October, Singapore, pp.1633–1636.

Yan, F., Kittler, J., Windridge, D., Christmas, W., Mikolajczyk, K., Cox, S. and Huang, Q. (2014) 'Automatic annotation of tennis games: an integration of audio, vision and learning', *Image and Vision Computing*, Vol. 32, No. 11, pp.896–903.