Noname manuscript No.
(will be inserted by the editor)

Accurate Ball Trajectory Tracking and 3D Visualization for Computer-Assisted Sports Broadcast

Mikel Labayen · Igor G. Olaizola · Naiara Aginako · Julian Florez

Received: date / Accepted: date

Abstract The application of computer-aided controversial play resolution in sport events significantly benefits organizers, referees and audience. Nowadays, especially in ball sports, very accurate technological solutions can be found. The main drawback of these systems is the need of complex and expensive hardware which makes them not affordable for less-known regional/traditional sports events. The lack of competitive systems with reduced hardware/software complexity and requirements motivates this research. Visual Analytics technologies permit system detecting the ball trajectory, solving with precision possible controversial plays. Ball is extracted from the video scene exploiting its shape features and velocity vector properties. Afterwards, its relative position to border line is calculated based on polynomial approximations. In order to enhance user visual experience, real-time rendering technologies are introduced to obtain virtual 3D reconstruction in quasi real-time. Comparing to other set ups, the main contribution of this work lays on the utilization of an unique camera per border line to extract 3D bounce point information. In addition, the system has no camera location/orientation limit, provided that line view is not occluded. Testing of the system has been done in real world scenarios, comparing the system output with referees' judgment. Visual results of the system have been broadcasted during Basque Pelota matches.

Keywords Computer Graphics · Camera Calibration · Tracking · Segmentation · Sports Events Broadcast

---

Mikel Labayen et all.
Department of Digital Television and Multimedia Services, Vicomtech - Ik4 Research Alliance
Tel.: +34-943 30 92 30
Fax: +34-943 30 92 30
E-mail: mlabayen@vicomtech.org,iolaizola@vicomtech.org, naginako@vicomtech.org, jflorez@vicomtech.org

1 Introduction

Object tracking has a prominent role within the field of computer vision. The proliferation of high performance computers, the availability of high quality video cameras at affordable prices, and the increasing need for automated video analysis has generated a great deal of interest in object tracking algorithms. Detection of target moving objects frame by frame, tracking and analysis to recognize their behavior are the usual pipeline in video analysis [2].

From application domain point of view, tracking systems are being introduced in sport game broadcasts, providing spectators with additional information. Due to the high performance equipment requirements, the renting of this kind of systems is quite expensive, making them unaffordable for small producers or broadcasters. This is exactly the Basque Pelota case. This regional/traditional game is produced by small producers and broadcasted by regional broadcasters. Their low budget does not allow to contract current setups to support controversial plays.

In this work, a system to assist referees solving controversial plays in sport games is described. The game has to be played with a ball and its playground must be delimited by lines. The developed software allows to reduce the set-up requirements, creating an accurate system that is affordable for a wider range of clients.

The set-up design, which is able to cover all border lines, is a challenge. This border line number can be high (e.g. tennis playground) driving the solution to multi-camera set-up. The modularity and scalability are important approaches for required solution. In this work The Basque Pelota test-case is presented in order to simplify the explanation. It is an ideal scenario to test the system first prototype because of its technical peculiarities. Since the playground is delimited by walls on 3 of its borders, it has only one border line to be covered. Once the application is validated in this game, this technology is being extended to other sports (i.e. tennis) which need a multi-camera distribution. For each camera image capture, image analysis and real-time rendering modules are reusable in this new modular and scalable set-up.

In the following subsection 1.1, this article carries out a short analysis of the state of the art in controversial play resolution. Afterwards, in chapter 2, a system overview is presented in terms of its objectives, description and specifications. Chapter 3 details the hardware and software (HW/SW) implementation of the core system, including camera calibration, image analysis and real time virtual 3D reconstruction processes. Finally, in chapter 4 the document shows the results obtained from tests carried out in real scenarios and it ends up with chapter 5 summarizing the conclusions.

1.1 Related Works

In its simplest form, tracking can be defined as the problem of estimating the trajectory of an object in the image plane as it moves around a scene. In other words, a tracker assigns consistent labels to the tracked objects in the different frames of a video. Additionally, depending on the tracking domain, a tracker can also provide object-centric information, such as orientation, area, or shape of an object. Therefore, the use of object tracking is pertinent in the tasks of [2]:

- Motion-based recognition, that is, human identification based on gait, automatic object detection, etc.
- Automated surveillance, that is, monitoring a scene to detect suspicious activities or unlikely events.
- Video indexing, that is, automatic annotation and retrieval of the videos in multimedia databases.
- Human-computer interaction, that is, gesture recognition, eye gaze tracking for data input to computers, etc.
- Traffic monitoring, that is, real-time gathering of traffic statistics to direct traffic flow.
- Vehicle navigation, that is, video-based path planning and obstacle avoidance capabilities.

This work focuses on motion-based object recognition in sport broadcasting. Tracking systems in the TV broadcast domain are not a recent approach at all. Most of the researched systems in this field are based on prediction algorithms based on Kalman [17] [4] or particle filters [10]. Extend state of the art material is available about methodologies dedicated e.g. to player and ball tracking in soccer [18] [14] [19] or tennis[7] [13] [5].

Some companies such as *Sportvision*[1] and *Virtual eye*[2] market systems which provide data content and enhancements for sports broadcasts and applications:

- The FoxTrax hockey puck tracking system [3] based on an infrared sensor. The circuit board inside a puck contained a shock sensor and infrared emitters. The puck emitted infrared pulses that were detected by both the 20 pulse detectors and the 10 modified IR cameras that were located in the rafters. Each IR camera processes the video locally and transmits the coordinates of candidate targets to the "Puck Truck".
- *Strick zone* control by ball and player tracking.Three PCs connected to three video cameras track a pitched baseball's flight toward the strike zone. Two cameras observe the baseball, while the third observes the batter to provide proper sizing for the strike zone.
- Playground lines drawings of *1ST & TEN*[3] in American football. This application uses a number of cameras shooting the field. Recent implementations require around four computers, one computer per camera plus a shared computer for chroma-keying and other tasks that can be run by a single operator.

[1] www.sportvision.com

[2] http://virtualeye.tv/

[3] http://www.ieeeghn.org/wiki/index.php/The_Making_of_Football\%27s_Yellow_First-and-Ten_Line

- Cricket[4] and golf ball tracking. Based on image computer graphics technology, 4 high-speed cameras (250 fps), two Infrared cameras and sophisticated computer rack are used to track the cricket ball. This set up needs at least a group of 4 operators to its management.
- Additional information for viewers as graphics and statistics in golf.

Due to their closed system, the algorithms on which they are based are in most of cases unknown.

Hawk-Eye[5] markets the most important controversial play image-based analysis and 3D virtual replay reconstruction approach for situations in which a tennis ball sized object is used in the play. Although, it started as cricket ball tracker, it is well known because it is able to point the location of a ball bounce in a tennis court with high accuracy. *Hawk-Eye* uses 6 high speed specialized vision processing cameras which are positioned around the ground and calibrated. In addition the system uses two broadcast cameras and calibrates them so that the graphic is always overlaid in the right place. All cameras have anti-wobble software to deal with camera movement. According to information in its web [9], it is able to deliver a pinpoint accuracy of under 5 mm.

However, the complexity of the set-up, high-speed cameras are needed, and the equipment requirements make the system too expensive for less-known regional/traditional sports. Even more, it cost is around $60.000 for one court which increases by 100 the cost of the system presented in this approach.

All these approaches use at least two or more high speed specialized vision processing cameras to determine the bounce distance from border line. In addition, they need operator team to control them. In order to reduce the existing solutions requirements, this work presents an alternative set-up, robust in terms of different possible camera operating location/orientation, based on unique broadcast-type camera per border line. This solution can be managed by one operator, even playground has more than one line under control, changing camera views from the system. The challenge in reduction of hardware complexity and achieving market solutions' accuracy motivates this work.

2 System Overview

The industrial project called *Begira*, in which this research has been carried out, establishes the technical specifications that the developed system has to fulfil. Although the state of the art can offer specific solutions for some of the technical requirements, it cannot afford the consecution of all the technical specifications. Even more, the economical limitations are a also the variables that constrains this research.

[4] `https://www.youtube.com/watch?v=LjLe06H7EJg`
[5] `www.hawkeyeinnovations.co.uk`

2.1 System Objectives

The system must be able to pinpoint accurately the distance of ball bounce from the line. The output 3D virtual video will simulate the last ball trajectory and will be inserted in TV PAL broadcast signal.The solution must be deployed on top of a simple HW/SW system to make it affordable for any producer, sport event broadcaster or less-known sport event organizer.

A system set-up design driven by flexibility in terms of size and operating location, can significantly reduce the costs rising this challenge as a major aim. In addition, the system needs to operate in quasi real-time, at least faster than the estimated time for video replay which is about 30 sec. Respect to pinpoint accuracy, defined by Basque Pelota referee committee, the estimation error should be under 1 cm for all cases and under 5 mm for 80% of them. This error has been set taking into account the typical human eye incertitude in the appreciation of bounce point (from a distance and with millisecond duration), which is also subjective. As this limit was considered achievable after the demonstration of our first version of the system, it was determined as a requirement.Referee committee is aware of the difficulty of approaching these accuracy values, but consider them necessary in order to standardize the system.

2.2 System Description an Requirements

In this section, the system general workflow, as well as module specifications and functionalities are described.

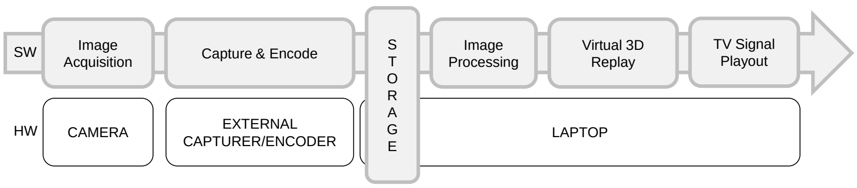

Fig. 1 System HW/SW Workflow & Modules

The prototype has four independent modules: a) the camera, b) the capturer/encoder, c) a laptop for storage and all processing tasks and d) a video adapter for TV PAL broadcast or playout. The camera captures images and transfers them in RAW format using an standard professional TV interface to the capturer/encoder module. This second module encodes frames using the H.264 codec and transfers it to the laptop where it stores them into a Transport Stream (.ts) video file. The image processing and later 3D virtual replay generation tasks are executed in the laptop. In the last module, the 3D virtual representation output video is adapted to broadcast quality video.

Table 1 Pixel/Distance(pixel/cm) ratios calculated for different image resolutions and camera distance

Image Resolutions	Distance from camera (m) 9 m	0.5 m	Reference Image
1080p	0.45	0.065	
720p	0.7	0.09	
576p	0.85	0.15	

- **Camera** The cameras used for image capture must fulfil some characteristics in order to make the ball detection easier for segmentation and identification operations carried out in the next steps:
  - **Frame Rate** The broadcast camera must provide enough images per second to track and predict the ball trajectory in each frame. The choice of this factor is defined taking into account the trade-off between the data processing time (in capturing/coding/storing) and the necessity of the amount of real images to be able to approach accurately the ball trajectory. To avoid missing frames, the time elapsed in recording/storing each frame must be less than reciprocal of the frame rate.
  - **Shutter speed and diaphragm aperture**
    In the case under study, the accuracy of the tracker can be improved if a target with stable shape, without blur effect, and with stable color (grayscale intensity) is acquired. Therefore, a high shutter speed camera is required. The choice of this factor is to be defined taking into account the trade-off between the minimum illumination required in the segmentation process and the necessity to keep the shape of the ball stable. It will be set to the minimum that allows the ball to appear as a clear round object. The maximum speed of the ball will be relevant on it (see Figure 2 and 3).
    The minimum shutter speed and consequently the diaphragm aperture are set according to the illumination of the sport events place and to the expectable maximum ball speed in each game. In this work, these values are set for Basque Pelota courts (indoor, illuminated for TV broadcast).
  - **Image resolution** The accuracy in ball bounce pinpointing is also related to the resolution of the captured images. The higher the resolution, the lower the pixels/distance ratio. Once again, the system performance is based on a trade off between lower processing time and higher accuracy in measure (see Table 1).
    To approximately calculate Pixel/Distance ratios, we use as reference object the border line. Calculating the amount of pixels in the horizontal and vertical vertices of the image, we can compute border line width pixel amount and compare it to border line real width measurement.
  - **Color space** The color space influences the ball segmentation process. Although multi-component color spaces can offer extra information in image object understanding, the bright white color of the ball and the

dark color of the playground, offer high contrast which makes single component color spaces enough for segmentation purposes. This reduces the generated data amount for capture, encoding, storage and processing tasks.
  - **Optical lens with fixed camera-to-playing field distance** The field of view of the camera must also be taken into account to determine lens distortion and system precision in terms of *image pixel/real distance*. The greater the field of view, the greater the covered scene area in which the ball trajectory can be analyzed. However, the greater the field of view, the greater the lens distortion and the lower the precision (pixel/distance). Even though this parameter must be taken into account, it is not as critical as others like velocity of the ball.
- **Capture & Encoding module** The capturer/encoder module captures the RAW multi component video signal provided by the camera. After that, the signal is converted to a single component color space. Then, the signal is compressed and encoded in order to reduce the information data flow for the storage and processing tasks.
  - **Codec** The codec requirements have to solve the controversial relation between image quality and compression ratio. The goal is to obtain the maximum compression ratio, keeping the minimum image quality which ensures correct segmentation conditions after decoding.
  - **File container** The video file must be read and written at the same time. In addition, the read process must offer quasi random access capabilities for the retrieval of part of the whole recorded video starting from a specific frame. Moreover, most of multimedia container formats include timestamps and data just before file closing, becoming navigation more difficult. The chosen container must solve this problem.
- **Storage and Processing Laptop** The laptop and capture/encoding module are connected trough USB 2.0. The laptop stores the encoded images in its hard disc, it retrieves and analyzes them and finally generates a 3D virtual replay of the action. A multi-tasking approach for quasi real-time performance establishes the hardware characteristics of the laptop. The system core software is stored and executed in this module. The algorithm robustness is directly related to the system set-up flexibility.
- **Video adapter to TV broadcast signal** The broadcasted output video signal must comply with the broadcaster graphical requirements and signal quality specifications at its mobile units. This module adapts the rendered video signal into a TV broadcast signal.

3 Implementation

In the first step of the implementation, state of the art and market study has been carried out to identify the existing HW/SW developments which best fit the needs of the system based on the requirements outlined above.

Two main issues have been encountered at this point: on one hand, no specific HW/SW solution exists for the established requirements. On the other, available HW solutions deal with independent tasks identified in the system workflow & modules figure, (Figure 1). This context pushes the development of our own algorithms, as the outcome of a research process. The unique existing Open Source algorithms used in the implementation are Camera Calibration (OpenCV) and Polynomial Approximation (GNU).

The system as a whole has been integrated using Qt[6]: a cross-platform application framework that is widely used for developing application software with graphical user interfaces (GUIs).

3.1 Image Capture

From the beginning, this system was developed using conventional TV broadcaster equipment in order to reduce costs in later market adoption processes. The camera used in the tested prototype is a common professional HD handheld camera (Panasonic HVX200A[7]), widely used by many kinds of producers/broadcasters. It provides a RAW YUV(4:2:2) component signal at a maximum resolution of FULL-HD 1080i and a maximum frame rate of 50fps far away from the throughput and features of cameras required by other market solutions. The camera is set-up at HD 720p 50fps both providing enough resolution and frame rate for our approach.

The scene illumination conditions are then to be analyzed. The amount of available light is a combination of pelota court lights and of additional spotlights used in special competition broadcasting. Under these conditions, the scene often is not enough illuminated, providing resulting images (at 50fps and 1/500 shutter speed) which are low-contrast.

Taking into account the minimum illumination required to keep the color and constant shape characteristics of the ball in the segmentation process, the balanced compromise between shutter speed, which keeps constant the ball round shape, and diaphragm aperture and electronic light gain, which keep the scene contrast, is defined for each broadcasted event.

Fig. 2 Low shutter speed (Auto). Electronic gain disabled. Blurry Ball

Fig. 3 High shutter speed (1/500). Electronic gain enabled. Blur-free Ball

[6] `www.qt.nokia.com`

[7] `www.panasonic.com/business/provideo/home.asp`

3.2 Image Encoding and Storage

The amount of data generated capturing HD 720p images at 50fps and described by the bit rate BR parameter makes necessary the use of compress/encode algorithms.

*Resolution:(1280 * 720) pixels/frame*
Frame Rate: 50 frames/sec
Bit Depth: 8 bits/pixel
Components: (1 (Y) + 0.5 (U) + 0.5 (V))
(Note: YUV 4:2:2 format)

$$BR = 1280 \times 720 \times 50 \times 8 \times 2 = 703.125 Gbps \quad (1)$$

The generated throughput would impose special storage, transmission bandwidth and equipment. This makes the system set-up more expensive and less compact. However, reduced system cost and dimensionality are central requirements from the beginning: to reduce the throughput the signal must be compressed. The dominant video codec today for web and mobile video (limited by the transmission channel bandwidth) is H.264 [15] [6]. H.264 compression preserves the video quality at high compression ratio better than other popular codecs widely available on the market [15] [16].

Although the standard defines 17 sets of profiles, H.264 has three commonly-used: Baseline (lowest), Main, and High. Higher profiles (Main and High profiles) ensure the best signal quality-compression relation. Since the system needs high compression ratios with the best signal quality, the High profile is chosen.

H.264 is typically deployed into *.MP4* file containers. However, a wide range of different containers can be used. One of the main difficulties of working with open videos is the random access within the content. Most seek function implementations require closed video files to function properly. However, in our case, the positioning at specific frame is performed will the video file is open and the encoder is appending information on it. To achieve this purpose, it is necessary to have time marks periodically embedded in the video file. Nevertheless, most video containers only include those marks just before the file is closed.

The container chosen to fit our requirements is therefore MPEG Transport Stream *(.Ts)* [12]. This Transport stream, devoted to content broadcasting, specifies a container format encapsulating packetized elementary streams, with error correction and stream synchronization features for maintaining transmission integrity when the signal is degraded. This allows to read specific video segments while writing into the same file.

The open source ffmpeg[8] library has been used to compress, encode and encapsulate the video, as well as to retrieve video sections, and decode them.

[8] `www.ffmpeg.org`

This package includes audio/video codec and audio/video container multiplexer and demultiplexer libraries.

3.3 Camera Calibration

Camera calibration or resectioning is the process of finding the true parameters of the camera that produced a given photograph or video based on prior knowledge of the scene. The camera parameters are classified in extrinsic and intrinsic parameters.

Rotation and translation matrices ($R, \overrightarrow{t}$) contain the extrinsic parameters which denote the coordinate system transformations from 3D world coordinates to camera coordinates. On the other hand, the intrinsic parameter matrix (K) encompasses focal length, image format, and principal point (Equation 2).

$$K = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \tag{2}$$

Where,
f_x & f_y - *Lens focal length*
c_x & c_y - *Principal point (the image center)*

The camera calibration is carried out using the so-called pinhole camera model, on which Opencv[9] camera calibration routines are based. A scene view is formed by projecting 3D points (x_p, y_p, z_p) into the image plane (x_i, y_i) using a perspective transformation.

$$P_i = \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix}, \quad P_p = \begin{pmatrix} x_p \\ y_p \\ z_p \end{pmatrix} \tag{3}$$

From the homography matrix (H), the matrices ($R, \overrightarrow{t}$) describing the rotation and translation parameters of the camera can be extracted.

In the mathematical development below the captured image points are identified by P_i (image coordinate, pixel) and playground plane points, where the ball will bounce, by P_p (real world plane coordinate, cm). P_p^* is an auxiliar point (real world plane coordinate, cm).

Since Opencv use homogeneous coordinates:

[9] `http://opencv.willowgarage.com/wiki/`

$$\begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} = K[R|t] \begin{pmatrix} x_p \\ y_p \\ z_p \\ 1 \end{pmatrix} \tag{4}$$

Where,
(x_p, y_p, z_p) *- Real world 3D coordinates*
(x_i, y_i) *- Projection point coordinates*

$$P_p^* = \begin{pmatrix} x_p^* \\ y_p^* \\ z_p^* \end{pmatrix} \tag{5}$$

$$H_{ip} = \begin{pmatrix} h_{(1,1)} & h_{(1,2)} & h_{(1,3)} \\ h_{(2,1)} & h_{(2,2)} & h_{(2,3)} \\ h_{(3,1)} & h_{(3,2)} & h_{(3,3)} \end{pmatrix} \tag{6}$$

$$P_p^* = H_{ip} \times P_i \tag{7}$$

$$H_{ip} = H_{pi}^{-1} \tag{8}$$

$$P_p = P_p^*/z_p^* = \begin{pmatrix} x_p^*/z_p^* \\ y_p^*/z_p^* \\ z_p^*/z_p^* \end{pmatrix} = \begin{pmatrix} x_p \\ y_p \\ 1 \end{pmatrix} \tag{9}$$

The homography matrix (H) needs to be calculated upon starting the system: it maps which pixels coordinates of captured image points P_i correspond with playground plane coordinate points P_p. Thus, once the bounce point in the captured image is identified, the position in the playground plane can be calculated.

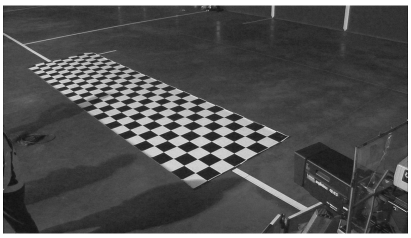

Fig. 4 Calibration checkerboard

The image points are selected using a calibration checkerboard as in Figure 4. The playground plane points are predefined and they must correspond to the points of the checkerboard which are selected in the captured image. With this process, camera intrinsic parameters matrix (K) and plane homography matrix (H) are calculated. Consequently, the (R, $\overrightarrow{t}$) matrices are defined.

The calibration information allows placing the camera (position and tilt) with respect to a reference point (x_p, y_p, z_p) of the pelota court world and determining its intrinsic distortion parameters. Undistort parameters and the geometric transformation, which establishes the relation between a captured image and the playground plane points parameters, are therefore established. Camera calibration makes the system robust in terms of different possible camera operating location. As a result, the system has no camera location/orientation limit, provided that line view is not occluded.

3.4 Image Analysis and Data Processing

In this section, the image processing and accurate bounce point determination algorithms are explained. The development has been based on the open source Opencv and GSL - GNU[10] libraries.

Once the system has been started, the recording begins. The camera is acquiring the contentious area around the playground border line and storing the information in a laptop where data is also processed during the entire duration of the game. The data captured from the camera is stored as MPEG transport stream (.ts) and using H264 encoding. When a controversial play occurs, the operator triggers the system. To that end, it extracts the latest frames, which contain the controversial play.

Once the set of images is extracted, the first image of the sequence is set as the background image (Figure 6). All the images of the sequence are converted from the color space, which is determined by the camera output, into a single component color space able to contrast the shape, movement and intensity descriptors to pinpoint the ball position in each frame.

After this, all the images are pre-processed using the camera intrinsic parameters matrix (K) to correct the distortion introduced by the optical lens.

After image preprocessing, the process for ball segmentation and tracking starts (see Figure 5) for each of the corrected images of the sequence (Figure 8). Due to the knowledge of the probable initial ball position, this process is only applied for a concrete image area. Image areas' difference is calculated pixel by pixel with respect to the previous image (temporally) (Figure 7) and to the reference image (Figure 6), such that two difference images (Figure 9 and 10 respectively) are obtained. Broadcast camera position and orientation are statics from the beginning of the match. For this reason, the frame difference technique provides a background-free output.

These two substraction image areas are transformed into black and white image areas via thresholding. The logic operation AND is performed for each

[10] `www.gnu.org/software/gsl/`

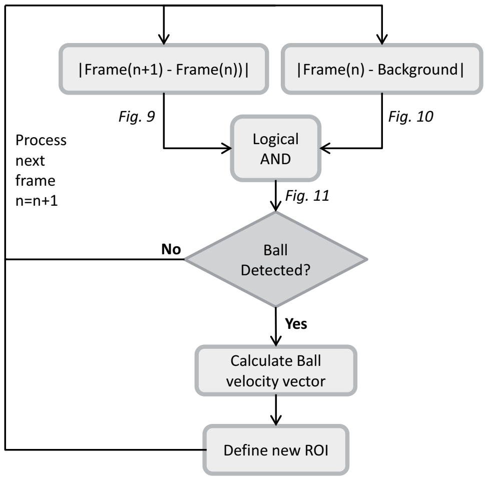

Fig. 5 Ball segmentation and tracking process

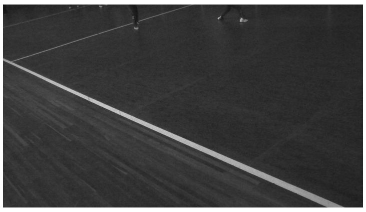

Fig. 6 Background reference image

Fig. 7 Extracted Frame(n)

Fig. 8 Frame(n+1)

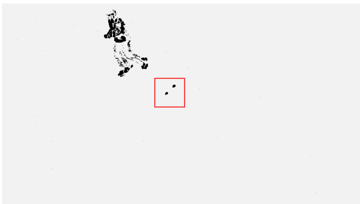

Fig. 9 |Frame(n+1) - Frame(n))|

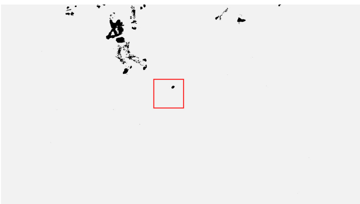

Fig. 10 |Frame(n) - Reference background|

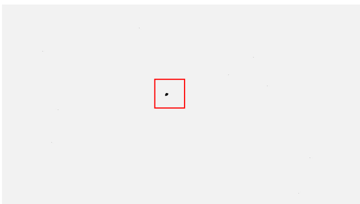

Fig. 11 Logical AND of (d) on (e) and resulting ROI

pair of image areas (Figure 11), so that only regions which are present in both images are extracted. Regions identified as noise also have to be removed by the logical AND composition operation. In order to discard noisy regions, estimated shape and area are used. Furthermore, velocity of the ball, considerably greater that of rest of the objects present in the scene, is set as key characteristic for segmentation. This methodology is used for extracting the initial position of the ball. Once the initial position is determined, the tracking process of the ball is performed.

The tracking process is based on the calculation of a movement vector. This movement vector and the velocity vector of the ball are calculated taking into account its coordinates (x_i''', y_i''') in pixels with respect to the previous image and to the time that has elapsed between one image and the next. In order to calculate the movement vector, the difference between the coordinates (x_i''', y_i''') of the center of the ball is calculated for consecutive images.

The calculation of the movement vector allows predefining a ROI where the segmentation process occurs. Once the initial point of the ball has been extracted and the movement vector calculated, the system creates a ROI determining the prediction area for ball position. All the process steps of frames substraction and AND logical operation will be made in the extracted ROI. Therefore, the process of ball detection speeds up.

The combination of camera position/orientation and selected ROI size keep usually the players belonging regions (noise) out of ROI. However, if there are more than one ball candidate region after AND composition operation, eliptic shape, calculated area and predicted position are used to discard irrelevant ones.

Tracking prediction algorithms, like Kalman Filter or Particle Filters, have been implemented and tested but finally rejected because they do not offer any significant improvements comparing with less complex procedure assuming some approximation. This is due to the fact that ball trajectory can be considered quasi linear close to bounce point. In addition, the relation between capture frame rate and ball velocity makes vector module almost constant and smaller than ROI size. For this reason, the velocity vector information is enough to predict properly the future ROI position and ROI size to detect the ball even if changes its direction after bounce.

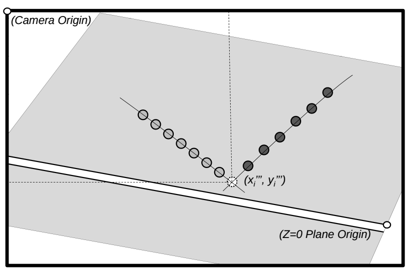

Fig. 12 Point Sequence split & polynomial approximation

As shown in Figure 12, once the velocity vector has been extracted for the entire sequence of frames, the sequence of positions of the ball (x_i''', y_i''') is divided into two segments. In order to define the limit of the segments, the difference in angle and modulus of the velocity vector is taken into account. The maximum value of the angle difference determines the limit which divides the two segments. If the angle values are similar, the modulus is used to break the deadlock.

Once the coordinates of the ball position of the ball have been determined for the two segments, a least-square fitting is performed for each of the two segments [1]. For the calculation of this fitting curve, the points which are above a minimum distance to the curve are iteratively discarded.

if,

$$|x_i'''(n) - lsf(x_i'''(n))| > \sum_{n=1}^{length} \frac{|x_i'''(n)-lsf(x_i'''(n))|}{length} \Rightarrow (x_i'''(n), y_i'''(n)) \text{ point discarded.} \quad (10)$$

Where,
lsf - Least Square Fitted function
length - Each segment length

The trajectory of the ball for each of the two segments is thus determined. The point of the intersection of the resulting curves is considered the bounce point of the ball(see Figure 13).

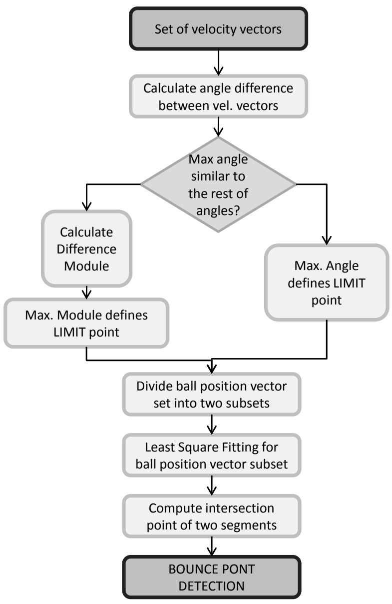

Fig. 13 Bounce point detection using velocity vectors

The position of the bounce point is now referenced with respect to the image coordinates (x_i''', y_i''') while the real distance of the bounce point to the playground border line is to be known. To that end, the geometric transformation obtained in the calibration procedure is applied to extract the coordinates (x_p, y_p) in the playground plane from the coordinates of captured image in pixels.

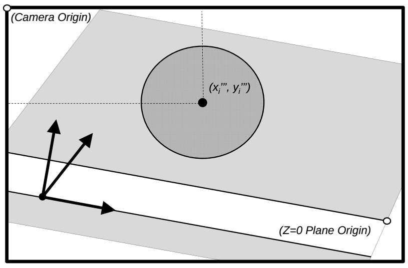

Fig. 14 The center of the ball (x_i''', y_i''') referenced to captured image origin

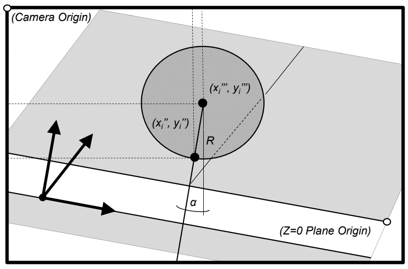

Fig. 15 The point (x_i'', y_i'') where the ball touch the ground referenced to image origin

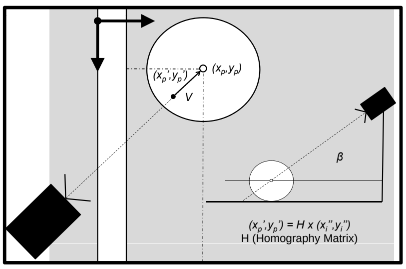

Fig. 16 Transformed touch point (x_p', y_p') at ground $(z_p = 0)$, referenced to ground plane origin

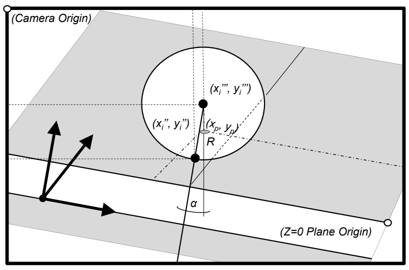

Fig. 17 Bounce point $(x_{p_{bounce}}, y_{p_{bounce}})$ referenced to ground plane origin

This geometric transformation, related to homography, can only be applied for the points of the playground plane $(z_p = 0)$. To that end, it is necessary to define the bounce point in the image. The point is defined by the coordinates (x”,y”) in pixels (Figure 15).

$$x_i'' = x_i''' + R * \sin(\alpha) \tag{11}$$

$$y_i'' = y_i''' + R * \cos(\alpha) \tag{12}$$

A small error is produced at the exact point where the ball touches the ground It has not been reflected in the figures, since the ball is superimposed.

$$x_p = x'_p + v_x(R, \overrightarrow{t}) = x_{p_{bounce}} \tag{13}$$
$$y_p = y'_p + v_y(R, \overrightarrow{t}) = y_{p_{bounce}} \tag{14}$$

Where,
$(x_{p_{bounce}}, y_{p_{bounce}})$ *- Bounce point at playground*

$$z_p = \begin{cases} (\frac{z_{p0}}{x_{p_{bounce}}} + \frac{g}{v_x^2} * (x_p - (2 * (x_p - x_{p_{bounce}})))) * (x_p - x_{p_{bounce}}) & \text{if } x_p < x_{p_b} \\ \frac{g}{v_x^2} * (x_{p_{bounce}}^2 - x_p^2) & \text{otherwise} \end{cases} \tag{15}$$

Where,
g - Gravity constant

Once the point (x''_i, y''_i) is calculated, the point on the real ground plane is obtained by multiplying it by the plane transformation matrix (H) (Figure 16 and Equation 7).

As seen in Figure 16, the point (x'_p, y'_p) is not an exact projection of the center of the ball, so it is moved in the direction of the optical vector of the camera with a distance which depends on the position and tilt of the camera to the central point.

Once the real position of the bounce point, which is referenced to the field line, has been calculated it can be determined whether the ball has bounced outside, inside or on the line itself.

Since the correct geometric transformation provided by the two plane homography only can determine the relation between captured images and playground plane points, the only actual ball 3D positioning can be carried out when it touches the ground (at bounce point). From this data, the rest of replay ball trajectory is simulated. Its (x_p, y_p) components are computed from the bounce point $(x_{p_{bounce}}, y_{p_{bounce}}, z_{p_{bounce}})$ and ball direction vectors defined taking into account some ball positions parameters in the processed images close to the bounce moment. The (z_p) component (See Equation 15) is based on parabolic model taking into account (x_p, y_p) points, approximate ball velocity vector and the z_{p_0} determined from the ball position at the first analyzed image frame.

3.5 Virtual 3D Replay

The visual result of the image analysis is the controversial play virtual 3D replay. Here one of the most performance demanding issues is the rendering engine, dedicated to the computational process of generating an image using 3D information. Firstly, this 3D shape information is converted into polygons and then into triangles. Secondly, these triangles are projected into a 2D image and, finally, each pixel inside the triangles is colored. The whole process takes too much time if no additional strategies or algorithms are used and live TV broadcast cannot be interrupted.

In order to address realtime 3D rendering, the approach is built on top of OpenSceneGraph[11] (OSG) library [11]. It is an open source, cross-platform graphics toolkit for the development of high performance graphic applications. It is based on the concept of a scene graph and uses OpenGL[12].

OpenSceneGraph makes use of techniques that speed up the rendering computational process because the rendering motor deals with considerably reduced information: a Level of Detail (LOD) algorithm, culling techniques (frustum, occlusion and small feature culling) and a State Sorting strategy are employed to this end.

The basic LOD idea is to use simpler versions of an object as it makes less and less of a contribution to the rendered image. So, when an object is far away, less polygons will be used to define it, which reduces the number of triangles to be processed in the rendering. The criteria OSG uses to select a level of detail model depends on the distance of the object from eye point (range-based selection). And to stop the switching form one LOS to another being noticeable, a Continuous Level of Detail (CLOD) technique is used.[8]

Culling techniques consist of removing portions of the scene that are not considered to contribute to the final image. The rest of the polygons are sent through the rendering pipeline. With the View frustum culling technique, all the polygon groups that are outside(the region of space in the modeled world visible form the eye point) are eliminated. When occlusion culling is used, all the objects hidden by groups of other objects are also eliminated from the sending-to-render process. And with Small Feature culling, small details that contribute little or nothing to the rendered images are not processed when the viewer is in motion.[8]

State Sorting consists of sorting geometrical shapes with similar states into bins to minimize state changes in the rendering process.[8]

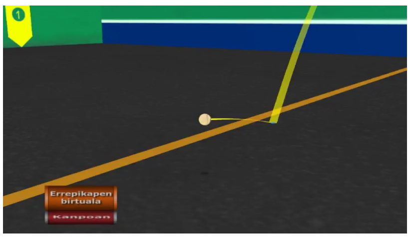

Fig. 18 3D virtual reconstruction of ball trajectory

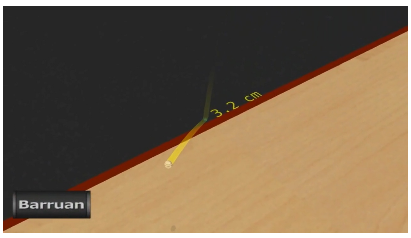

Fig. 19 Virtual representation of bounce point

The 3D visualization module takes as input the ball 3D trajectory, the exact bounce point, its distance from the line, its shape and if the bounce point is

[11] `www.openscenegraph.org`
[12] `www.opengl.org`

in or *out*. According to this incoming data, the scenario is loaded and the ball trajectory simulated creating an output video file with the controversial play reconstruction. Although the output video is rendered by a conventional camera view for standard TV broadcast, it can be rendered with stereoscopic cameras for future 3DTV broadcasts.

The module configuration defines the variable parameters which describe the scenario: playground border lines width and color, ground material texture, rendered measure number and arrow colors, etc. This makes it easier to configure the virtual scenario for the different real pelota courts where the game takes place.

3.6 Signal Adaptation for PAL-Quality digital TV signal playout

With regards to the output signal, the system is able to provide HD 1080p digital video throughput. However, it is required to also be compatible with nowadays Standard Definition (SD) broadcaster TV signal standards. Rendered images are adapted to these restrictions. The output video is rendered with a Matrox4 CG2000[13] video adaptor, since this hardware combines a 3D graphic accelerator with broadcast quality video I/O.

The system output signal can be adapted to lower quality formats (PAL 4:3, 16:9) if it is needed due to compatibility issues.

4 System evaluation

The initial assessment phase of the parameters described in Section 2.2 specifies in depth which parameters of the camera improve the further segmentation process.

According to this first assessment, the scene often is not illuminated enough for proper image acquisition. To improve the image quality, the balanced compromise between shutter speed, diaphragm aperture and electronic gain is set. Furthermore, the thresholds used in ball-player-background segmentation are set according to camera parameters and playground illumination. The evaluation of the system has been done before a professional TV broadcasted Basque Pelota match in three different stadiums. In sport broadcast, the courts must be well illuminated in order to maximize the contrast between foreground and background and make the ball visible over the playground for the audience and TV viewers. Accordingly, the illumination and its changes are under control during match time and our system take advantage of this stable environment.

The test set-up has been another considerable challenge. The ball physic behavior has been studied and tested, reaching that the ball bounce can be considered elastic, because its hardness, ruling out any deformation on its shape or track. In order to get to the conclusion that bounce area remains circular, tracing paper has been used. This tracing paper is set along the border

[13] `www.matrox.com`

Table 2 Accuracy test results for tests made in Ogueta(Vitoria) playground

	Test 1			Test 2		
Distance from camera [d] (m)	**Real measure (cm)**	**System measure (cm)**	**Error (cm)**	**Real measure (cm)**	**System measure (cm)**	**Error (cm)**
d < 1.5 m	8.40	8.65	0.25	-8.20	-8.34	0.14
d < 1.5 m	2.85	2.86	0.01	7.10	6.60	0.50
d < 1.5 m	3.75	4.12	0.37	0.80	0.57	0.23
1.5 m < d ≤ 3 m	0.75	-0.04	0.79	-0.50	-0.90	0.40
1.5 m < d ≤ 3 m	-3.75	-3.95	0.20	-5.20	-5.80	0.60
1.5 m < d ≤ 3 m	-0.10	-0.34	0.24	-7.80	-7.56	0.24
3 m < d ≤ 4.5 m	19.00	18.85	0.15	-3.00	-2.58	0.42
3 m < d ≤ 4.5 m	1.75	0.90	0.85	-0.70	-1.07	0.37
3 m < d ≤ 4.5 m	-3.20	-3.15	0.05	-3.55	-2.90	0.65
4.5 m < d ≤ 6 m	3.70	2.45	1.25	-4.75	-3.95	0.80
4.5 m < d ≤ 6 m	6.50	5.64	0.86	2.50	3.40	0.90
4.5 m < d ≤ 6 m	2.00	1.40	0.60	-6.20	-5.80	0.40
6 m < d ≤ 7.5 m	7.10	7.37	0.27	4.00	4.70	0.70
6 m < d ≤ 7.5 m	6.90	6.95	0.05	3.20	3.85	0.65
6 m < d ≤ 7.5 m	-1.70	-1.75	0.05	2.70	3.10	0.40
7.5 m < d ≤ 9 m	3.30	3.50	0.20	1.90	2.30	0.40
7.5 m < d ≤ 9 m	4.20	4.50	0.30	12.60	12.00	0.60
7.5 m < d ≤ 9 m	11.90	12.00	0.10	2.10	1.80	0.30

line in order to determine real distance measurements during testing period. Millimetric paper is set under the tracing paper to get the real distance from the bounce area center and the border line. Consequently, the real distance measurement error can not be above 0,5mm to achieve market competence.

As mentioned before, the system has been tested in different playgrounds. For all tests, the selected parameters for the system are:

1. **Frame rate** = 50fps
2. **Shutter speed** = 1/500sec
3. **Diaphragm aperture** = 1:1,7
4. **Image resolution** = 720p HD
5. **Camera lens focal length** = 35mm
6. **ROI window size** = 100X100px
7. **Detection color space** = Gray scale
8. **File storage codec** = H.264

The camera was located 3 meters from the ground and on one side of the line. The field of view allowed covering 9 meters (as shown in Figure 20), enough to cover the controversial action play zone. The camera pan and tilt were different for each test, determined to each playground. The system can be adapted for any kind of sport event, taking into account that the error can vary depending the referees requirements.

According to the results extracted from the tests made in one of the playgrounds(see Table 2), the algorithm robustness is proved along different pos-

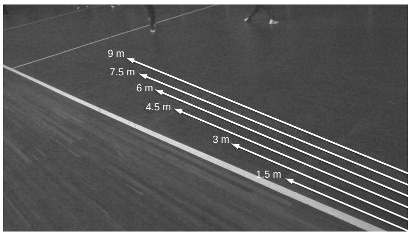

Fig. 20 Covered distances in the testing trials

sible camera operating distances. Average error for both test measurements is 4,3mm, which is below the target 5mm deviation. The numerical error requirements listed in Section 2.1 are not fulfilled successfully, because the 80% of measures should be below this 5mm error threshold and only the 69,4% of the errors satisfy this requirement. still, 80% of the errors are below 7mm.

Although Referee Committee considers this results as acceptable because the typical human eye incertitude in the appreciation from a distance, with millisecond duration, is considerably higher than that of the presented system, ongoing research is being developed in order to fulfill these requirements. One of the objectives of this ongoing research lays on the use of FullHD cameras instead of HD cameras. These cameras represent the same scene using a greater number of pixels and therefore the pixel-real distance ratio decreases, permitting a more precise calibration process and the minimization of error (in real cm measumerent) when ball centre point is detected. The other major issue comes from the control of playground illumination, in order to improve the ball segmentation process.

The tests reveal that the precision in measurements is related to the accuracy in ball center pointing (in each captured frame) and to the accuracy in the homography matrix calculation, both of which are closely related to image resolution. Actually, the error in measurement is not constant across the field. Although the resolution of the image is constant, the real distance that a pixel represents (pixel/distance ratio) is different depending on camera location. The longer the distance between the line-point and the camera, the lower the (pixel/distance) ratio. Nevertheless, the experimental results show that this theoretical issue is not crucial for distances less than 9 m from the camera at HD 720p resolution.

In the springs of 2010, 2011 and 2012 the system was tested in the most important Basque Pelota competitions. Although the numerical measurements did not accomplish the goals of the Referee Committee, they considered the system ready to help them taking decisions during the match. The system worked as expected on professional platforms and the output signal was broadcasted live by the Basque public broadcaster (EiTB[14]) successfully. In 2010 it was watched by 219.000 spectators and the viewer share was 31,1%.

[14] www.eitb.com

The opportunity of broadcasting the virtual 3D repetition of the bounce permits to the spectator to get more information about the ongoing match. Due to the velocity of the ball and the limits of the broadcasting cameras, it's no viable to reproduce the last recorded frames and detect the bounce point of the ball. Only making a reconstruction of the followed track it's possible to determine this point. Therefore, the user experience is enhanced using the results of the described system.

5 Conclusions

In this work a low-cost automatic ball bounce detector and 3D virtual replay generator is proposed for sport event broadcast. The central engineering trade-off choice approach has been to reduce the bounce detection system set-up and hardware requirements to unique broadcast-type camera per border line as well as to reduce the system software to quasi real-time performance. The challenge of hardware complexity reduction keeping accuracy in results can be considered the main technical contribution of this work.

The algorithm introduces an additional advantage which makes it more flexible in terms of different possible camera operating location. Contrary to other approaches, the camera and the playground plane can form any angle, since the necessary transformations for calculating the point of impact in real coordinates are effective and accurate. For this reason, the system has no camera location constraint, provided that line view is not occluded.

The typical human eye incertitude in the appreciation of distant actions(a few meters from linesman to bounce events point), with millisecond duration (because of the ball speed) is considerably higher than the score of the presented system. Obtained results show that the measure errors are close to the demanded range in order to standardize the system for Basque Pelota events.

The use of 3D virtual reality for controversial action replays in sport event broadcasting enhances audiences and TV spectators' visual experience. Due to the reduction of the production costs, this contribution represents a new opportunity for less-known traditional/regional sport events to use this technology, as well as, for small producers, organizers and broadcasters to compete with well-known competition organizers and expensive broadcasting rights owners.

The experience of having developed a research effort applied to real world deployment for sports events has materialized a complete solution covering he whole production chain. Technical specifications and hardware requirements for a system that has to be included in a real world implementation are stronger than the ones required for a system with not so close relation with real world applications. Even more, several variables are no more under control of the researcher, which makes the work harder.

This work has been granted with the patent EP2455911 **Method for detecting the point of impact of a ball in sports events**

Acknowledgements The authors would like to acknowledge the collaboration offered by G93 Telecomunicaciones[15] (Audio-Visual, Computer and Graphic Services for Television) and EiTB (Basque public broadcaster) for the help offered in the system development, test and broadcast processes.

The authors are also grateful for the collaboration offered by ASPE[16] (ASPE Jugadores de Pelota) in providing access to its professional pelota player training sessions and in advising in game rule issues as well as for the financial support offered by research project programs of the SPRI[17] (Society for Industrial Promotion and Restructuring of Basque Country).

Finally, the authors would like to thank the rest of *Begira* research team: Maider Laka, Julen García and Aritz Legarretaetxebarria. Also, Javier Barandiaran and Iñigo Barandiaran for their advice and the collegues of *Digital Television and Multimedia Services* department for the unconditional help offered.

References

1. Ahn, S.: Least squares orthogonal distance fitting of curves and surfaces in space. Lecture notes in computer science. Springer (2004). URL `http://books.google.es/books?id=we4cHJBFzLwC`
2. Alper Yilmaz, O.J., Shah, M.: Object tracking: A survey
3. Cavallaro, R.: The foxtrax hockey puck tracking system. Computer Graphics and Applications, IEEE **17** (1997)
4. Erik Cueva, D.Z., Rojas, R.: Kalman filter for vision tracking
5. F.Yan, W., J.Kittler: A tennis ball tracking algorithm for automatic annotation of tennis match
6. Gary J. Sullivan, P.T., Luthra, A.: The h.264/avc advanced video coding standard: Overview and introduction to the fidelity range extensions. SPIE Conference on Applications of Digital Image Processing XXVII, Special Session on Advances in the New Emerging Standard: H.264/AVC (2004)
7. Gopal Pingali Agata Opalach, Y.J.: Ball tracking and virtual replays for innovative tennis broadcasts
8. Haines, E., Akenine-Moller, T.: Real-Time Rendering, 2 edn. AKPeters (2002)
9. Innovations, H.E.: Hawk-eye accuracy and believability (www.hawkeyeinnovations.co.uk). URL `http://www.hawkeyeinnovations.co.uk/UserFiles/File/Hawk-Eye%20accuracy%20and%20believability2.pdf`
10. Isard, M., Blake, A.: Condensation - conditional density propagation for visual tracking
11. Laka, M., Garcia, I., Macia, I., Ugarte, A.: Tv sport broadcasts: Real time virtual representation in 3d terrain models. In: 3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video, 2008. IEEE eXplore, pp. 405–408. Istanbul, Turkey (2008)
12. Miller, F., Vandome, A., McBrewster, J.: MPEG Transport Stream. VDM Publishing House Ltd. (2010). URL `http://books.google.es/books?id=tkxvcAAACAAJ`
13. N. Owens C Harris, C.S.: Hawk-eye tennis system
14. Naidoo, W.C., Tapamo, J.R.: Soccer video analysis by ball, player and referee tracking
15. Richardson, I.E.: The H.264 Advanced Video Compression Standard, second edn. Vcodex Limited, UK (2003)
16. Richardson, I.E.G.: Video Codec Design: Developing Image and Video Compression Systems. Jhon Wiley & Sons (2002)
17. Wu, W.: Tennis touching point detection based on high speed camera and kalman filter
18. Xinguo Yu Hon Wai Leong, C.X., Tian, Q.: Trajectory-based ball detection and tracking in broadcast soccer video
19. Xinguo Yu Changsheng Xu, H.W.L.Q.T.Q.T., Wan, K.W.: Trajectory-based ball detection and tracking with applications to semantic analysis of broadcast soccer video

[15] `www.g93.es`
[16] `www.aspepelota.com`
[17] `www.spri.es`