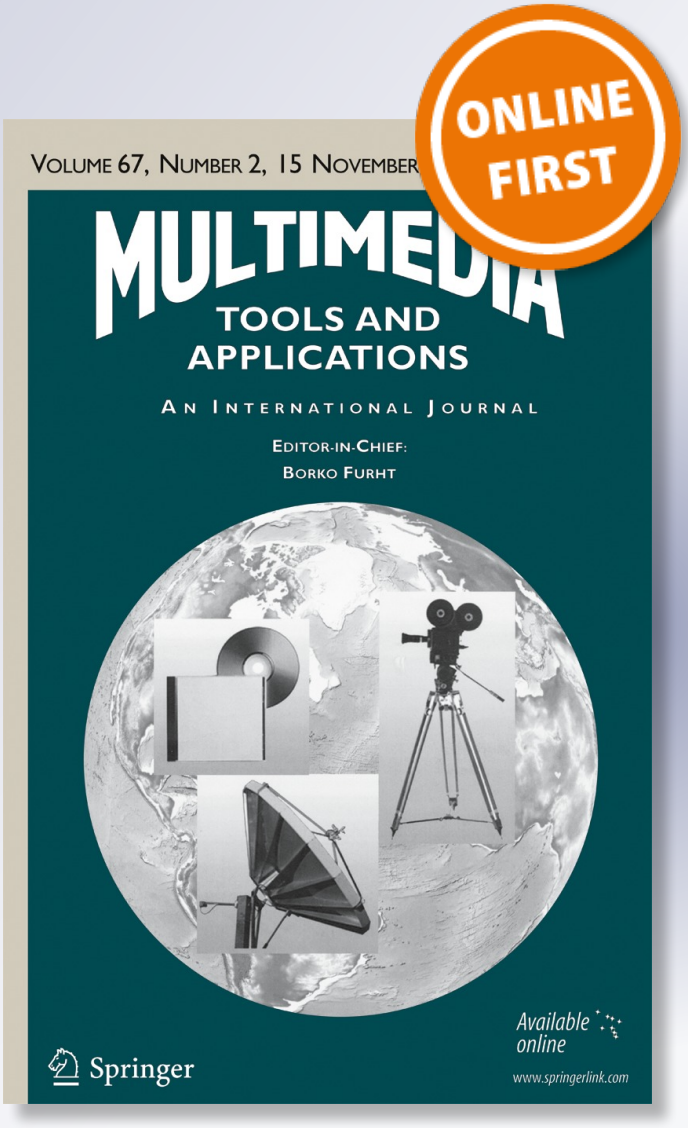
A semi-supervised system for players detection and tracking in multi-camera soccer videos

Rafael Martín & José M. Martínez

Multimedia Tools and Applications
An International Journal

ISSN 1380-7501

Multimed Tools Appl
DOI 10.1007/s11042-013-1659-6



Springer

Multimed Tools Appl
DOI 10.1007/s11042-013-1659-6

A semi-supervised system for players detection and tracking in multi-camera soccer videos

Rafael Martín · José M. Martínez

Abstract This paper presents a complete, general and modular system which after a simple previous configuration is able to detect and track each player on the court or field. The presented multi-camera system is based on a mono-camera object detection and tracking system originally designed for video surveillance applications. Target sports of the developed system are team sports (e.g., basketball, soccer). The main objective of this paper is to present a semi-supervised system able to detect and track the players in multi-camera sports videos, focusing on the fusion of different tracks of detected blobs in order to match tracks across cameras. The proposed system is simpler than other systems from the state of the art, can operate in real time and has margin to be improved and to reduce supervision adding additional complexity. In addition to the detection and tracking system, an evaluation system has been designed to obtain quantitative results of the system performance.

Keywords Sports videos · Multi-camera systems · Object detection · Object tracking · Fusion · Homography

1 Introduction

Sports broadcasts constitute a major percentage of the total of public and commercial television broadcasts. A lot of work has already been carried out on content analysis of sports videos and the work on enhancement and enrichment of sports video is growing quickly due to the great demands of customers.

An overview of sports video research is given in [24], describing both basic algorithmic techniques and applications. Sports video research can be classified into the following two main goals: indexing and retrieval systems (based on high-level semantic queries) and augmented reality presentation (to present additional information and to provide new viewing experiences to the users). The analysis of events can be carried out by combining various attributes of the video, including its structure, events and other content properties. In

R. Martín (✉) · J. M. Martínez
VPULab, EPS – Universidad Autónoma de Madrid, Madrid, Spain
e-mail: rafael.martinn@estudiante.uam.es

J. M. Martínez
e-mail: josem.martinez@uam.es

Published online: 20 August 2013

event detection, features can be extracted from three information channels: video, audio and text. The definition of what is interesting differs for each viewer. While sport fans are more interested in events like spectacular goals, the coaches might be more interested in the errors of the players to help them to improve their capabilities.

In addition, algorithmic building blocks must consider structure analysis (of a match), object detection and segmentation (e.g., position of players, shape and movements of the athletes), ball and player tracking (specially the tracking of a small ball is a difficult problem), camera calibration (to determine the geometric mapping between the image coordinates and the real world positions for, among others, 3D reconstruction), etc.

There are two main types of sports video, used to perform the analysis:

- *Edited broadcast videos* [8, 10, 18, 25]: videos are edited for broadcasting on television. On this type of videos, there are scenes of the match, repetitions and changes of viewpoint of the observer. This type of videos is the most common available.
- *Multi-camera videos*: In general, multi-camera videos are not edited. They are subdivided into two main classes:
  - *Mobile* [14]: Cameramen follow the players, changing the orientation of their cameras. The obtained videos usually are used to generate the broadcast videos, mixing and editing parts of them.
  - *Fixed* [3, 23]: The cameras remain fixed from the beginning to the end of the recording. In this case, no operator is needed for controlling the orientation of the camera.

The presented system is centred on team sports. We use the term "team sports" for sports where two or more players move in the same area of the field. Referees may also be moving in the same areas and among the players. The main objective of this paper is to present a semi-supervised system able to detect and track the players in multi-camera sports videos, focusing on the fusion of different tracks of detected blobs in order to match tracks across cameras. This work is based on a mono-camera detection and tracking system designed to work for video surveillance applications and that has been adapted to operate within the sports domain.

For team sports fusion is harder than for individual sports [15], because all the tracked blobs do not belong to the same player. During tracking, occlusions may happen causing problems in tracking, like losing players or generating fused blobs from more than one player. Some examples of this kind of sports are soccer, basketball or volleyball.

The remainder of the paper is structured as follows: after this introduction, section 2 presents a brief overview of the related work; Section 3 describes the designed system, including the fusion method; Section 4 depicts the evaluation system; Section 5 presents the adjustments, testing and results of the implementation of the system, including the description of the used dataset; Section 6 shows some applications based on the detection and tracking results of the system; finally, Section 7 describes the conclusions and future work.

2 State of the art

The canonical system for detecting and tracking players in a field can be decomposed into several main blocks, as depicted in Fig. 1. The different techniques and associated algorithms that are part of any sports video analysis system are described below.

Background extraction is a simple and very common method used for moving objects segmentation which consists of the difference between a set of images and the background

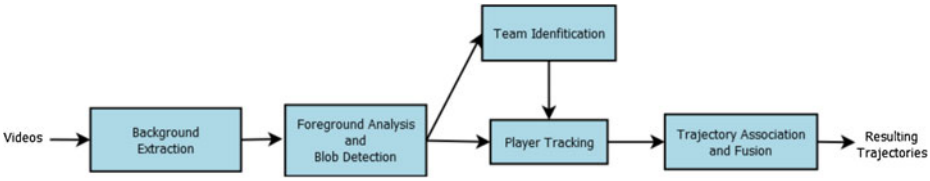


Fig. 1 Canonical system for sports video analysis

model. The background model is generated from an empty image of the field (if it is available) or from a fragment of the video with non-static players. Some of the main problems in background extraction are the changes in the environment, such as illumination, shadows or background moving objects. Some statistical adaptive methods are used in [8] for this block. A histogram learning technique is employed in [10] to detect the playfield pixels, using a training set of soccer videos.

In [25], background is modelled with a mixture of Gaussians and learned before a match without the need of an empty scene. A background detector is first used to extract a pitch mask for saving processing time and avoiding false alarms from the crowd [14]. Proposes a background model and a background updating procedure which is used to continuously adapt the model to the variations in illumination conditions. In [3], the region of the ground field is extracted by segmenting out non-field regions like advertisements and afterwards the players are extracted on this field on the basis of the field extraction. The field is represented in [23] by a constant mean colour value that is obtained through prior statistics over a large data set. The colour distance between a pixel and the mean value of the field is used to determine whether it belongs to field or not. In the system, only hue and saturation components in the HSV colour space are computed to exclude the effect of illumination changes.

Blob detection is based on a ***foreground analysis*** to identify the players. The goal is to adjust the detection, as much as possible, to the contour of the player blob, separating the near or overlapping possible players. In [8], the nodes (blobs) of a graph are grouped considering the weights of the edges (distances) among them. The area of the blobs is the parameter used to define the number of components of each node. The graph is constructed from the set of blobs obtained during the segmentation step in such a way that nodes represent blobs and weights of edges represent the distance between these blobs. In [10], connected component analysis (CCA) scans the binary mask and groups pixels into components based on pixel connectivity. The foreground mask is morphologically filtered (closing after opening) in [18] to get rid of outliers. This is followed by a connected component analysis that allows creating individual blobs and associated bounding boxes. In [25], for an isolated player, the image measurement comes from the bottom of the foreground region directly. The measurement covariance in an image plane is assumed to be a constant and diagonal matrix, because foreground detection is a pixel-wise operation. A further step with the connectivity analysis has been introduced in [14] to extract connected regions and remove small regions due to noise. This procedure scans the entire image and groups neighbour pixels into regions. The connectivity analysis eliminates shadows by using geometrical considerations about the shape of each region.

The information of the uniform of the teams is used for identifying the ***team*** and differencing among players, goalkeepers or referees. Generally a player can be modelled as a group of many regions, each region having some predominant colours.

Dividing the model of the player in two or more regions is attempted in [8], so that each region represents a part of the team uniform (e.g., t-shirt, short, socks). For each region, a filtering based on the vertical intensity distribution of the blobs is defined.

Colour histograms are used to classify the objects of templates for each class in [18]. The different classes consist of team 1 player, team 2 player, team 1 goalkeeper, team 2 goalkeeper and referee. The centre part of the detected blobs is taken and colour histograms are calculated. Histograms are compared, after normalization, with the colour histograms of the templates using the Bhattacharyya distance. This block can be also implemented using a histogram-intersection method [25]. The result for each player is a five-element vector, indicating the probability that the observed player is wearing one of the five categories of uniform (the categories are the same as described in [18]). In [3], the group behaviour of soccer players is analysed using a colour histogram back-projection to isolate players on each team. But as different teams can have a similar histogram, vertical distribution of colours is used.

When the position of each player position is detected, the next objective is to get the player tracking to know where the player is at each moment. In [8], the tracking of each player is performed by searching an optimal path in the graph defined during blob detection. At each step, the minimal path in the graph is considered, using the distance information between the blobs. The bounding box and centroid coordinates of each player are used in [25] as state and measurement variables in a Kalman filter. Finally, in the track selection there is a procedure of tracking aided recognition for the 25 most likely players (11 from each team and 3 referees). Due to false alarms and tracking errors, there normally exist more than 25 tracks for the players. A player likelihood measure is calculated for each target on the basis of the confidence of the category estimate, number of support cameras, domain knowledge in positions (for goalkeepers and linesmen), frames of being tracked or missing, as well as the fixed population constraint. A fast sub-optimal search method gives reasonable results.

The multiplayer tracker may use the Maximum a Posteriori Probability [14] to get the best fit of state and observation sequences. The state vector includes information about the location, velocity, acceleration and dimension of the single bounding box. Template matching and Kalman filter are used in [3]. The templates of players are extracted from the player mask using connected component analysis. First, new players that do not significantly overlap with the bounding box of a player already tracked are found out. Then, the new players are inserted to the tracking list. The location of players at the next frame is predicted by Kalman filter and Template matching at that location is performed. Finally, the player template is updated. The main problem of player tracking is occlusion and in [3] only occlusion between different teams is considered. In [2], starting from a frame where no two players occlude each other, the players are tracked automatically by considering their 3D world coordinates. When a merging of two (or more) players is detected, the separation is done by means of a correlation-based template matching algorithm.

The ***trajectory association*** across the multiple points of view requires the ***fusion*** of multiple simultaneous views of the same object as well as the fusion of trajectory fragments captured by individual sensors. There are two main approaches for generating the fused trajectories: fuse and then track [22], and track and then fuse [1, 13, 14, 21, 25].

A multi-camera multi-target track-before-detect (TBD) particle filter that uses mean-shift clustering is presented in [22].

The information from multiple cameras is first fused to obtain a detection volume using a multi-layer homography. To track multiple objects in the detection volume, unlike traditional tracking algorithms that incorporate the measurement at the likelihood level, TBD uses the signal intensity in the state representation. The association matrix also can be decided according to the Mahalanobis distance [25] between the measurement coordinates and the track prediction. To solve the correspondence problem, a graph based algorithm can be applied [14]. The best set of tracks is computed by finding the maximum weight path. In

[13], after obtaining the transformed trajectories, the next step is to compute their relative pair-wise similarities for association and fusion in order to have a single trajectory corresponding to an object across the entire field. The parameters used in [1] for the association are: shape, length, average target velocity, sharpness of turns (which defines the statistical directional characteristics of a trajectory), trajectory mean and variation information at the sample level (using PCA). With all those parameters, the parameter vector is generated and the cross correlation is used as proximity measure. The problem of association is posed and solved in a graph-theoretic framework in [21].

Multi camera analysis algorithms that generate global trajectories may be separated in three main classes, namely, appearance based [3, 12, 17], geometry based [5, 9, 11, 19] and hybrid approaches [6, 7, 16, 21]. Following [13], the main ideas of each class are described: appearance-based approaches use colour to match objects across cameras; geometry-based approaches establish correspondences between objects appearing simultaneously in different views; hybrid methods use multiple features to integrate the information in the camera network.

Table 1 presents an overview of all the reviewed sport video analysis techniques.

3 Video analysis system

3.1 System overview

In this section the system is presented. The system block diagram is depicted in Fig. 2.

3.2 Initialization

The initialization of the system can be obtained from a previous video recorded before the match, ideally when the players are warming up. If this previous video is not available, the first frames of the match video can be used for this initialization.

The initialization step generates the field representation, the backgrounds, the masks, the homography reference points and the thresholds for the fusion. The actual measures of the field are needed for the field representation. The background of each camera can be generated with any state of the art methods. Masks are easily defined by regions defined by points selected by the system user, as well as the homography reference points.

3.3 Mono-camera detection and tracking

The mono-camera detection and tracking (mcD&T) module is based on a video analysis system for abandoned and stolen object detection [20]. For validation, we also make use of a "perfect" detection and tracking module, named ground truth detection and tracking (gtD&T).

This system is designed to work as part of a video-surveillance framework capable of triggering alarms for detecting events in real time. This requirement imposes limits on the time complexity of the algorithms used in each of the analysis modules.

After the initial frame acquisition stage, a foreground mask is generated for each incoming frame at the *Foreground Segmentation Module*. This foreground mask consists on a binary image that identifies the pixels that belong to moving or stationary blobs. Then, post-processing techniques are applied to this foreground mask in order to remove noisy artefacts and shadows. After that, the *Blob Extraction Module* determines the connected

Table 1 Overview of sport video analysis techniques

Reference	Sport	Background extraction	Foreground analysis and blob detection	Team identification	Player tracking	Trajectory association and fusion
[8]	Football	Statistical adaptive methods, Gaussian distribution	Graph representation	Vertical intensity distribution	Optimal path in the graph	–
[10]	Football	Learning histogram	Connected component analysis	–	–	–
[18]	Football	–	Morphological filters, connected component analysis	Colour histogram contrast	–	–
[25]	Football	Mixture of Gaussians	Measurement covariance	Histogram intersection	Kalman filter	Mahalanobis distance
[14]	Football	Background with updating procedure	Connectivity analysis	–	Maximum a Posteriori Probability	Graph based algorithm, Hopkroft and Karp algorithm
[3]	Football	Peak values of histogram, morphological filtering	–	Vertical distribution of colours	Template matching, Kalman filter	–
[23]	Football	Mean colour, colour distance	–	–	–	–
[2]	Football	–	–	–	Template matching	–
[22]	Basketball	–	–	–	–	Particle filter, multi–layer homography, K-means, mixture of Gaussian clustering
[13]	Football	–	–	–	–	Feature vector correlation
[1]	Basketball and football	–	–	–	–	Feature vector correlation
[21]	–	–	–	–	–	Graph based algorithm

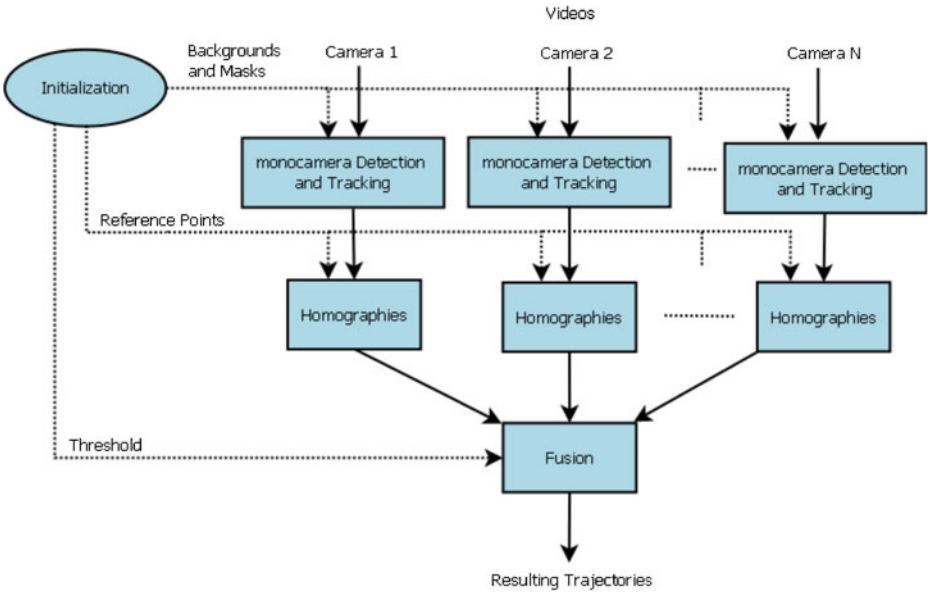


Fig. 2 System block diagram for team sports

components of the foreground mask. In the following stage, the *Blob Tracking Module* tries to associate an ID for each extracted blob across the frame sequence.

Figure 3 depicts the block diagram of the mono-camera detection and tracking module.

3.3.1 Foreground segmentation module

The purpose of the *Foreground Segmentation Module* is the generation of binary masks that represent whether pixels belong to the background or foreground (moving or stationary blobs).

Based on the BackGround Subtraction (BGS) segmentation technique, a background model is created and then updated with the incoming frames. This initial mask then undergoes noise and shadow removal operations in order to obtain the final foreground mask for the current frame; finally connected component analysis if performed for blob extraction.

The background model is adaptively updated to consider slow changes in global illumination conditions using a running average method. Then, the distance to the background model is calculated for each incoming frame.

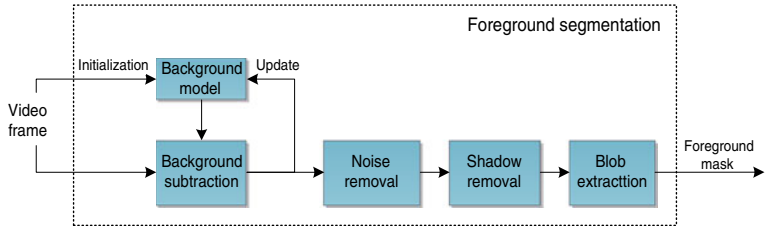


Fig. 3 Block diagram of the mono-camera Detection and Tracking Module

A shadow removal technique is applied to the foreground mask for removing those pixels that belong to shadows produced by moving or stationary entities (e.g., objects and people). For this purpose, the Hue-Saturation-Value (HSV) colour space is used, as it allows us to explicitly separate between chromaticity and intensity.

Additionally, morphological operations are performed on the resulting foreground mask for removing noisy artefacts. In particular, a combination of erosion and reconstruction operations, known as "Opening by Reconstruction of Erosion", is applied. Its purpose is to remove small objects (in our case blobs due to noise), while retaining the shape and size of all other blobs in the foreground mask.

After applying the background subtraction and post-processing the obtained foreground mask, the *Blob Extraction Module* labels each isolated groups of pixels in the mask using Connected Component Analysis. The implemented algorithm uses 8-connectivity as the criteria to determine if pixels belong to the same connected region.

3.3.2 Blob tracking

This module performs tracking of the blobs extracted by the previous module. This is done by estimating the correspondence of blobs between consecutive frames (current and previous frames). A Match-Matrix (MM) is used to quantify the likelihood of correspondence between the blobs in the previous and current frame. For each pair of blobs, the values of this matrix are computed using the normalized Euclidean Distance between their blob centroids and their colour. It is calculated as follows:

$$MM_{nm} = \sqrt{\left(\frac{\Delta X}{X\mathrm{dim}}\right)^2 + \left(\frac{\Delta Y}{Y\mathrm{dim}}\right)^2} + \sqrt{\left(\frac{\Delta R}{255}\right)^2 + \left(\frac{\Delta G}{255}\right)^2 + \left(\frac{\Delta B}{255}\right)^2} \qquad (1)$$

where each row n corresponds to blobs in the previous frame and each column m to the number of blobs in the current frame; ΔX and ΔY are the differences in the X and Y directions between the centroids in the past and previous frames, normalized to their maximum values (the frame dimensions X dim and Y dim); ΔR, ΔG and ΔB are the differences between the mean R, G and B colour values, also normalized to the maximum value (255 for 8-bit RGB images).

Then, the correspondence for each blob m is defined as the index i (i=1…n) that presents the minimum value MM_{im}.

3.4 Homographies

A homography is an invertible transformation from a projective space (for example, the real projective plane) to itself that maps straight lines to straight lines and points to points.

It describes what happens to the perceived positions of observed objects when the point of view of the observer changes. Projective transformations do not preserve sizes or angles.

The code used for the homographies is public[1] and follows the normalized direct linear transformation algorithm given by Hartley and Zisserman [9]. Four points are selected in the image plane and their correspondences are indicated in the top view. The grass mowing pattern facilitates the choice of the points.

In Fig. 4, an example of the result is shown. Note that in the system the homography is applied to a single point (the base mid-point) for each player tracked. The red points in the figure indicate selected points.

[1] http://www.csse.uwa.edu.au/~pk/research/matlabfns/

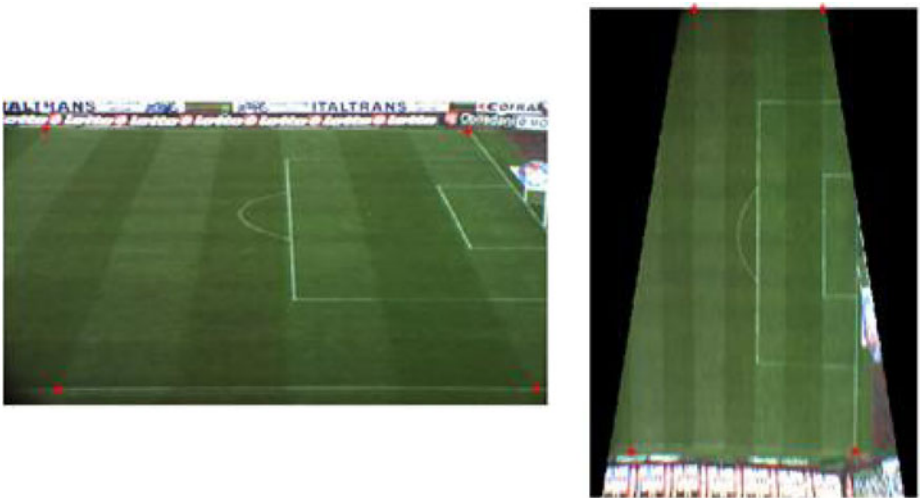


Fig. 4 Example of homography: **a** image plane, **b** Top view

Resulting trajectories from two opposite trackings may have location differences due to scale changes (large area coverage), players volumes, camera distances (different sizes), lens distortion, etc. In the case of people detection and tracking, these differences use to be moderate and, therefore, the fusion block will make use of accurate enough trajectories.

In our case, the differences between the first two pairs of cameras (cameras 1 to 4) are acceptable. In the case of the pair of cameras 5–6, the error detected is higher. Therefore, it is corrected as detailed in section 5.1.2.

3.5 Fusion

For the fusion process, the following information is obtained for each blob: number of frames, initial frame, final frame, number of frames and coordinates (x and y) for each frame.

A threshold is defined to decide which blobs should be fused. All the scores under the threshold indicate that both blobs belong to the same player. The method to calculate the score is explained in this section. Each score is calculated from two blobs, one from each tracking.[2] When we refer to 'two trackings' in this section, we refer to the resulting tracking from two different cameras. It should not be confused with the two available types of tracking (ground truth tracking, gtD&T system, and the tracking system referred as mono-camera detection and tracking system, mcD&T system). Each one of the two blobs belongs to a different tracking. The two trackings must have an overlapping field zone for the fusion. This threshold can be calculated from a training sequence with the evaluation system presented in section 4.

A training video can be used (for example, in the warm up period previous to the match) in which an operator can adjust the thresholds by selecting them interactively.

When a score is obtained for each pair of blobs from two different trackings, the next step is to obtain a list of blobs associations (LOA) containing one row for each blob of one of the

[2] In this case the tracking is defined as the set of blobs that is fused with another set of blobs

trackings. Each row indicates which blobs in the other tracking should be fused with the blob that corresponds to that row. A similar LOA for the other tracking is not necessary to calculate because it is redundant, although sometimes it is useful for simplifying further processing (e.g. in the step where the resulting blobs of the fusion process are calculated). To obtain the LOA, row by row, all the blobs of the second tracking that should be fused with the blob corresponding to the row (from the first tracking) are added in consecutive columns. Table 2 shows an example a LOA.

The information extracted from the figure is: blob 1 of the first tracking is fused with blobs 5 and 15 of the second tracking (if there were more blobs fused, more columns will appear); blob 2 of the first tracking is fused with blob number 20 of the second tracking; blob 4 of the first tracking is fused with blob number 21 of the second tracking; blob 5 of the first tracking is fused with blobs 4 and 19 of the second tracking; blobs 3 and 6 do not fuse with any blob.

Finally, for the fusion, the LOA is processed to get the group of blobs in each camera that should be fused.

The proposed fusion approaches are described below. The fusions use only information of the position of each player. The fusion improvement adjusts the spatial projection of the trajectories of each player.

Given two blobs, the score is defined as the mean square distance per frame, for those frames in which both blobs appear.

$$Score(blobI, blobJ) = \\ = \begin{cases} \frac{1}{N}\sum_{f=n}^{m}\left(\left(x_{f,i}-x_{f,j}\right)^2+\left(y_{f,i}-y_{f,j}\right)^2\right), & \text{If there are frames in which both blobs appear} \\ \infty\ , & \text{Else} \end{cases} \quad (2)$$

where $x_{f,i}$ and $y_{f,i}$ are the coordinates of the blob I in the frame f; $x_{f,j}$ and $y_{f,j}$ are the coordinates of the blob J in the frame f; n…m are the frames where blob I and blob J are tracked simultaneously; and N is the number of frames in which both blobs appear (m-n+1). The blob coordinates are the coordinates corresponding to the base mid-point of the bounding box of the player.

4 Evaluation system

For the evaluation system, in addition to the data described in the fusion section, the ground truth unique ID of the blob is needed, corresponding to the player or referee. This ID is

Table 2 Example of LOA

	Blobs to fuse from the second tracking		
		1	2
Blobs from the first tracking	1	5	15
	2	20	0
	3	0	0
	4	21	0
	5	4	19
	6	0	0

obtained from the ground truth.[3] This ID allows knowing if two blobs belong to the same player and evaluating how many fusions are correct or erroneous. The unique ID is only used for the evaluation system. The ID that presents the tracking system for each blob of each camera has no relationship with this unique ID.

Two blobs should be fused if they belong to the same player (both have the same unique ID) and if there are frames in which both blobs appear. For the fusion of two trackings, a list like the LOA described in section 3.5 is obtained, but with the certainty that in this case the ideal list of blobs associations (iLOA[4]) is obtained, with all the fusions that should ideally be done. This iLOA allows calculating the Precision and Recall (defined at the end of this section) when it is compared with the experimental list of blobs associations (eLOA).

After obtaining the iLOA and defining a set of values for the threshold defined in Section 3.5, eLOAs are obtained. Each eLOA is compared to the iLOA, obtaining the successful and wrong fusions in each case.

If a blob number is found in the same line in both lists, the fusion is correct. The total number of elements in the iLOA corresponds to the total number of expected fusions. The total number of elements in the eLOA corresponds to the total number of fusions obtained by the system.

After obtaining the necessary data, the values of Recall and Precision for the fusion process are calculated: Recall is the fraction of accurate associations to the true number of associations; Precision is the fraction of accurate associations to the total number of achieved associations. Let ξ_Ω be the ground truth for pairs of trajectories on the overlapping region and let E_Ω be the estimated results. Then, Recall and Precision are calculated as:

$$Recall = \frac{|\xi_\Omega \cap E_\Omega|}{|\xi_\Omega|} \tag{3}$$

$$Precision = \frac{|\xi_\Omega \cap E_\Omega|}{|E_\Omega|} \tag{4}$$

5 Adjustments, testing and results[5]

5.1 Content set: ISSIA soccer dataset

For the testing, the used videos are from ISSIA Soccer Dataset [4]. This public dataset[6] is composed of:

- Six synchronized views acquired by six Full-HD cameras (DALSA 25-2M30 cameras), three for each major side of the playing-field, at 25 fps (6 AVI files).

[3] In the ISSIA soccer dataset, this ID is extracted from the ground truth tracking of each of the 6 cameras. The blob ID for the blobs of a player is the same for the six tracking files.

[4] The LOA, iLOA and eLOA are "classes", which are instantiated as specific lists (e.g., FCeLOA, RiLOA) during the different fusions performed (see section 5.2).

[5] A web page has been created where some videos with the result of the system have been published: http://www-vpu.eps.uam.es/publications/ASemiSupervisedSystemForPlayersDetectionAndTrackingInMulticameraSoccerVideos/

[6] http://www.issia.cnr.it/htdocs%20nuovo/progetti/bari/soccerdataset.html

- Manually annotated objects position of 2 min of the match. These metadata provide the positions of the players, referees and ball in each frame of each camera (6 XML files). The players have the same labels while they move in the six views that correspond to the numbers on their uniforms. The player labels of the first team start from 1 while the player labels of the second teams start from 201.

  The first 300 frames of each sequence have not been labelled in order to provide an initial phase to initialize the background subtraction algorithms.
- Calibration data in the form of pictures containing some reference points in to the playing- field and the relative measures for calibrating each camera into a common world coordinate system (6 pdf files).

The positions of the six cameras on the two sides of the field are shown in Fig. 5.

The available ground truth tracking consists of an ideal tracking of each camera in which, besides the position and size of each player in each frame, the unique ID of each blob is provided. It allows knowing which player corresponds to each blob. This tracking had some annotation errors that were corrected when they were detected.[7]

Other camera layouts may be used, but depending on the layout characteristics the results may be affected. Taking into account the characteristics of the ISSIA datasets, the results will vary depending on:

- The availability of multiple cameras in the ISSIA dataset, covering the same areas, allows reducing errors if one tracker loses the player; if the dataset lacks of this redundancy, the system will have more errors as some players tracked by a unique camera may be lost during some frames.
- The availability of facing cameras in the ISSIA dataset allows to solve occlusions; if the dataset lacks facing cameras, the system will have more errors (to be supervised manually at the moment). Future improvements in the mono-camera detection and tracking (mcD&T) module will reduce those errors.
- The existence of overlapping regions in the ISSIA dataset allows to solve additional (to the ones solved by the facing cameras) occlusions at the regions frontiers and to facilitate regions fusion; if the dataset lacks overlapping regions, there will be more errors (to be supervised manually at the moment). Additional solutions will include doing the region fusion using prediction for players exiting a region towards an adjacent one without overlapping frames.
- The limitation in the variation of players size (due to the similar location of the cameras in the ISSIA dataset) facilitates the tracking process as the size of the searched players is less variable; if the dataset lacks of this location characteristic, the system will have more errors because the tracking module will get worse results. Enhancements in the mono-camera detection and tracking (mcD&T) module will reduce those errors.

5.1.1 Definitions

There are some important definitions that facilitate understanding the rest of the section:

- Facing cameras: overlapping cameras covering an area of the field. There are three pairs of facing cameras: camera 1 and camera 2, camera 3 and camera 4, and camera 5 and camera 6.

[7] A document with the corrections is available in the created web page.

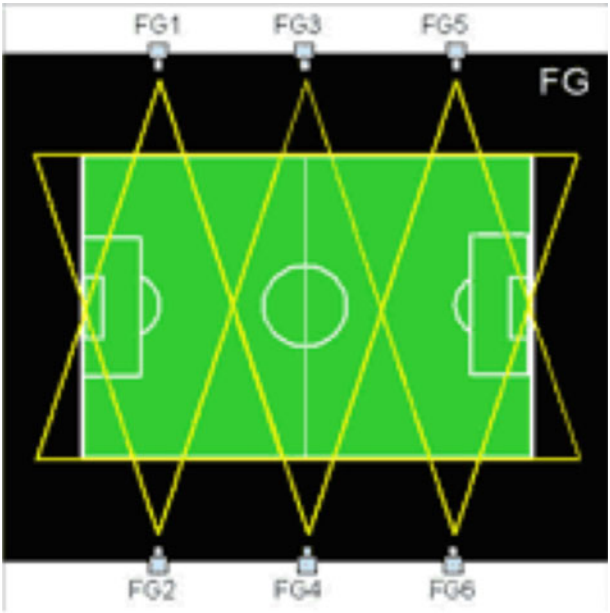


Fig. 5 Positions of the six cameras of ISSIA Soccer Dataset (Extracted from http://www.issia.cnr.it/htdocs%20nuovo/progetti/bari/soccerdataset.html)

- Regions: resulting regions from the fusion of each pair of facing cameras. There are three regions, one for each pair of facing cameras.
- Field: result of combining the three regions, covering all the field area, by fusing the two overlapping areas of the regions (region of cameras 1–2 and region of cameras 3–4, and region of cameras 3–4 and region of cameras 5–6).

To get the fusion of the field, first facing cameras are fused and then the resulting regions are fused, generating the fusion of the field.

The number of correct fusions for each fusion ground truth is represented in the Table 3.

The fusion threshold takes values between 0 and 50. The approach developed for fusion is explained in section 3.5.

There are two different evaluations of the system used, one for each available tracking, which are described in sections 5.3 and 5.4.

For each of the two available tracking data (the one provided in the ground truth of individual cameras and the one resulting from the tracking of the proposed system) the result of the fusion described in section 3.5 has been analysed.

Table 3 Number of correct fusions

Tracking	Camera 1 with 2	Camera 3 with 4	Camera 5 with 6	Total facing cameras fusions	Regions fusion
Ground truth tracking	23	75	39	137	106
mcD&T system	159	462	267	888	132

5.1.2 Homography adjustments

Fusion between cameras 5 and 6 shows results which are worse than that in other pairs, because the homography does not fit properly due to the reasons discussed in section 3.4: imperfections of the lens and different camera orientation and height. In Fig. 6, an example of the problem is shown. Two fragments of the resulting trajectories are presented from the player with unique ID 104 of the facing cameras 1 and 2 (right) and 3 and 4 (left). The vertical distance of the trajectories from cameras 5 and 6 is significantly higher than the distance between the trajectories from cameras 1 and 2.

The presented results use the fusion after correcting the explained error. The example shown in Fig. 6 with the correction of the points of the homography is presented in Fig. 7.

The correction is done heuristically and manually after seeing the results.

In a real system, the correction would be done at initialization. From a warming up video with players moving around the field, the trajectories of players are extracted and the correct fit can be obtained.

As this correction is not compensated in the other facing cameras, the results of regions fusion are slightly worse, but the goal is to show that the correction of the homography can significantly improve the results of the corrected trajectories.

5.2 Steps followed for each validation scenario

In this section, the steps for obtaining the fusion of the field, resulting in the trajectories of players, and getting the data needed (instances of LOAs) for the evaluation of the system are presented.

Steps for obtaining the trajectories of players (fusion of the field):

1- The annotated blobs are obtained for the different frames after the tracking. In the case of the ground truth tracking, the result is directly the ground truth tracking file, and for the mcD&T tracking, the result is the generated output file of the mcD&T module.

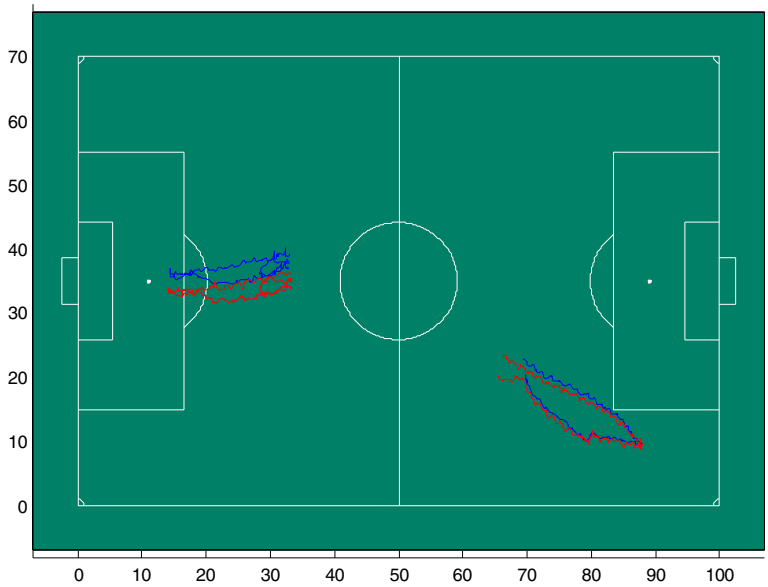


Fig. 6 Example of trajectories before applying the homography without correction

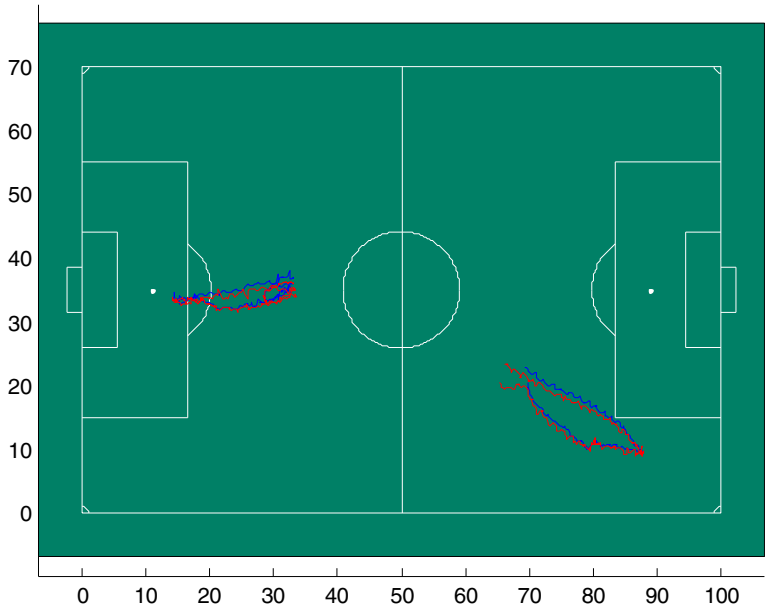


Fig. 7 Example of trajectories after applying the homography with correction

2- The homography corresponding to each of the camera tracking is applied to obtain the top view trajectories of each camera tracking.
3- The first *experimental Lists of Blobs Associations* (eLOAs) are calculated between each two groups of blobs belonging to facing cameras, resulting in *Facing Cameras experimental Lists of Blobs Associations* (FCeLOAs). There are 3 fusions in total resulting in 3 FCeLOAs, one for each pair of facing cameras.
4- The facing cameras blobs are fused according to the obtained fusion lists (FCeLOAs). After this step, the 6 trackings are reduced to 3 trackings, one for each region.
5- The second fusion lists of blobs associations are calculated between the groups of blobs of the different overlapping regions (region of cameras 1–2 and region of cameras 3–4, and region of cameras 3–4 and region of cameras 5–6. See Fig. 8 to see an example with the resulting overlapping areas), resulting in *Regions experimental Lists of Blobs Associations* (ReLOAs). There are 2 fusions in total resulting in 2 ReLOAs.
6- The blobs of the different regions are fused according to the calculated ReLOAs. After this step, the final field tracking with all the trajectories is obtained (see Fig. 12 to see an example of the resulting field trajectories).

Steps for obtaining the results of the evaluation system:

1- Making use of the FCeLOAs, the evaluation of the facing cameras is made. The ideal lists (FCiLOAs, different for each used tracking because they depend on the resulting tracking blobs) are calculated with the unique ID (the way to get the unique ID is explained in each scenario: in the ground truth tracking it is the true ID contained in the tracking files, and in the mcD&T module it is obtained with spatial and temporal similarity between blobs of the mcD&T module and the ground truth) and compared with FCeLOAs (the lists obtained with the different threshold and fusions), obtaining the Precision and Recall values for facing cameras.

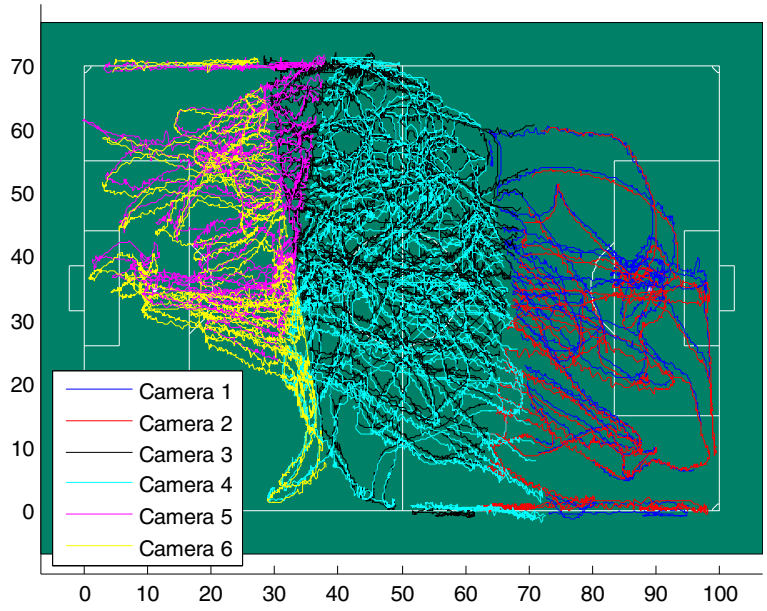


Fig. 8 Top view tracking for each camera

2- For the evaluation, in the fourth step for obtaining the regions fusion, the fusion ground truth (ideal list) is used to prevent that the errors in this first stage affect in the evaluation of the next stage. If the ideal fusion is not applied for facing cameras fusion, the second evaluation cannot be calculated. A unique ID cannot be assigned to the resulting blob of the fusion of blobs from different players. Furthermore, when two blobs belonging to a player are not fused in the first stage, but are fused in the second stage, two correct fusions are obtained instead of one, which changes the final results. Note that in the process for obtaining the trajectories of the players (without evaluation) the ground truth is not used.
3- Step 1 for obtaining the results of the evaluation system is repeated, but in this case with the RiLOAs and ReLOAs. The results obtained are the Precision and Recall values for the region fusion.

5.3 Validation scenario 1: gtD&T

The first testing uses as tracking results the ground truth tracking files provided in the dataset. There are six files, one for each camera, containing the tracking for each player. The blob ID for the blobs of a player is the same for the six tracking files and is called unique ID. For example, the blobs of the player with unique ID 5 have that identifier for all the tracking files. The unique blob ID is not used to facilitate the fusions; it is only used to evaluate, obtaining the Precision and Recall values as described in section 4. Thus ground truth tracking is used as if it had been obtained from a "perfect" tracking system with the advantage that quantitative results of performance can be obtained.

The first step is to apply the homography corresponding to each of the six tracking files to obtain the top view trajectories. The result is shown in Fig. 8.

Then, the facing cameras fusion is calculated. Results from the fusion between facing cameras are shown in Fig. 9, which is obtained joining the results obtained from all the

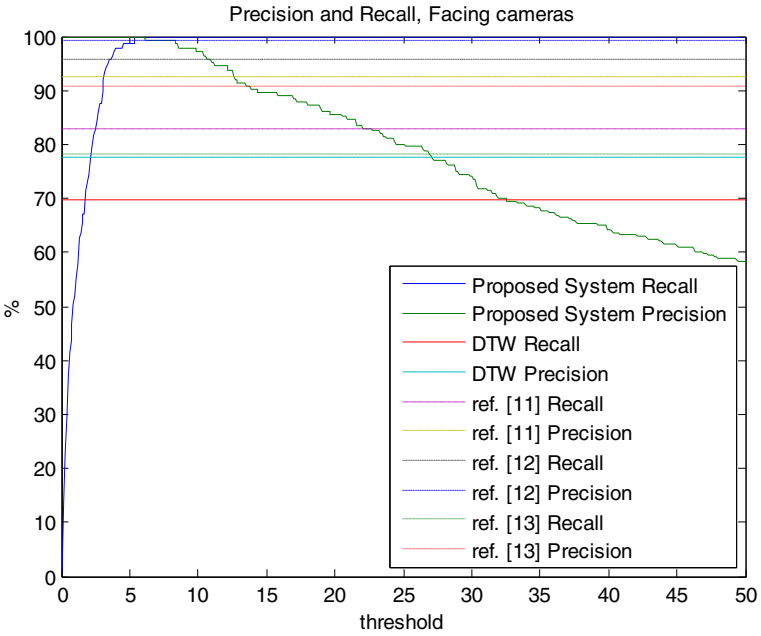


Fig. 9 Precision and Recall from fusion of facing cameras

facing cameras fusion. The result is not the average from the 3 pairs because the number of fusions in each pair of cameras is not the same.

The ideal result of the facing cameras fusion is presented in Fig. 10. In the figure, each colour (red, green and blue) represents a region.

Finally, the regions fusion is calculated. Figure 11 shows the results of fusing the three regions.

The ideal result of the field fusion is presented in Fig. 12. In the figure, each player is represented with a different colour.

5.4 Validation scenario 2: mcD&T

The second testing uses the tracking module of the mcD&T system. The average time of the tracking processing of the videos (with the applied modifications to the mcD&T module) is 7,7 frames per second.

The first step is to apply the homography corresponding to each of the six ground truth tracking files and to each of the six mcD&T module tracking files to obtain the top view trajectories of both trackings.

For the evaluation of the system, the unique IDs obtained by the mcD&T system are mapped to the ground truth IDs. To get the unique ID of the mcD&T module tracked blobs, the score defined in section 3.5 between the top view ground truth tracking blobs and the top view mcD&T module tracked blobs is calculated for each camera. For each blob of the mcD&T module, the corresponding blob of the ground truth tracking with the lowest score indicates the unique ID. Using this method, the identifier of the closest spatially blob (in the corresponding frames) from the ground truth tracking is obtained for each of the blobs obtained from the mcD&T module tracking.

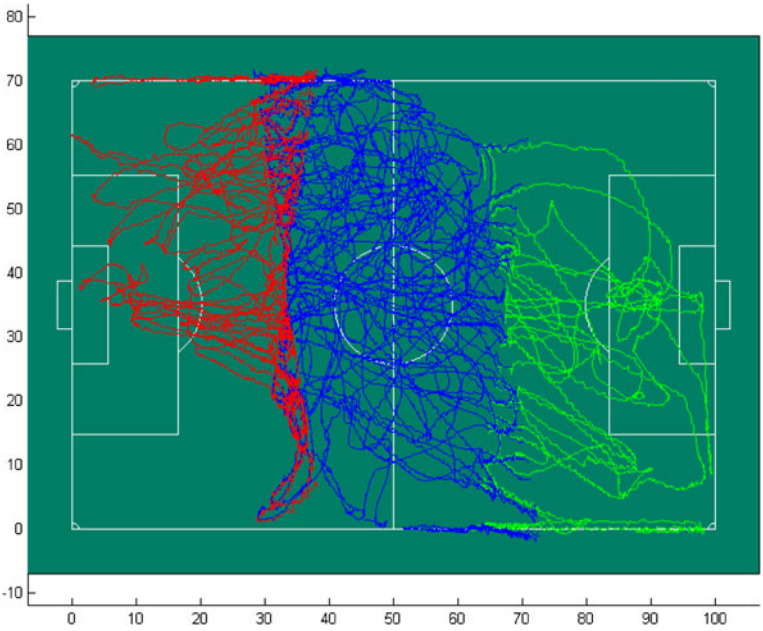


Fig. 10 Resulting trackings from the fusion of facing cameras

After obtaining the unique ID of the blobs of each camera of the mcD&T module tracking, the steps 3 to 6 of section 5.2 are applied: FCeLOAs are obtained, the blobs of the facing cameras are fused according to the FCeLOAs, ReLOAs are obtained and, finally,

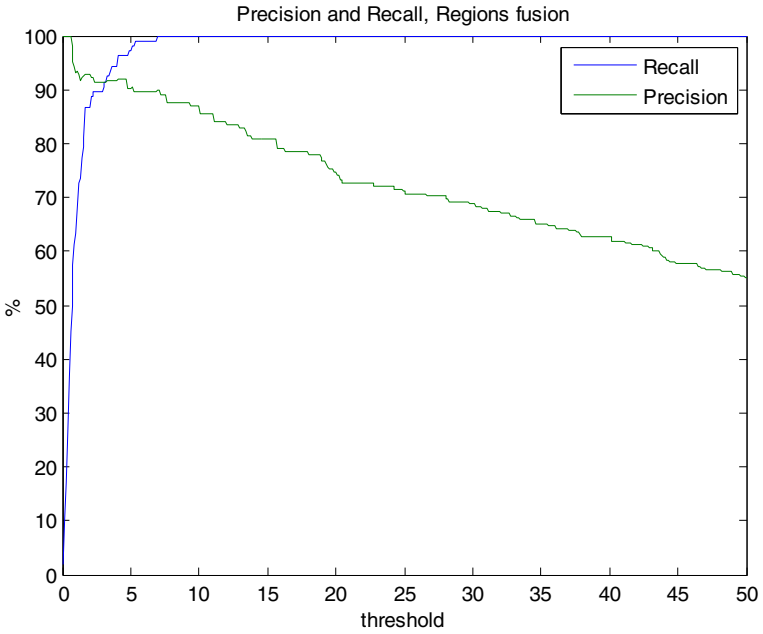


Fig. 11 Precision and Recall from fusion of the different resulting regions with Ground Truth tracking

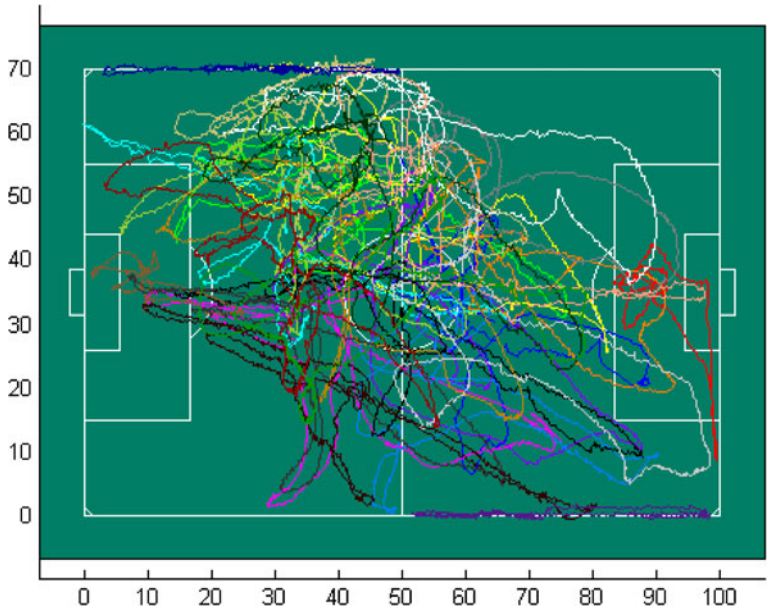


Fig. 12 Resulting tracking from the region fusion

the blobs of the different regions are fused according to the ReLOAs, resulting in the field trajectories. As in the case of the previous scenario, the unique blob ID is used only to evaluate Precision and Recall after the fusion, as described in section 4.

The Precision and Recall values for this validation scenario are shown in the following figures. Figure 13 is obtained joining the results obtained from all the facing cameras fusion.

Figure 14 shows the results of fusing the three regions.

5.5 Computational performance

A time analysis of the process is presented in Table 4. The execution has been done on a computer with an Intel Core I7@2.66GHz and 6 GB of RAM memory. Note that these times are calculated on a Matlab implementation which is not optimized to reduce its running time. The presented times are given for the complete sequence. The presented times are given for the complete sequence. The average processing time, assuming parallelism in the processing of each camera, is 837 s, that is, the processing speed is about 3.3 frames per second. An efficient real implementation could reach real-time.

As can be seen in the results, as expected, the execution time depends on the number of blobs for fusion. In the case of the fusion between cameras 3 and 4, the execution time is higher than for the other two pairs, as the number of blobs is also higher. The execution time of the regions fusion is the greatest of all because in this fusion all the resulting blobs from the fusion of the three pairs of cameras are processed.

5.6 Results comparison

For the comparison of results, we use the equal error rate point as our resulting value. So using the ground truth tracking, we get a 100 % Precision and Recall using a threshold value

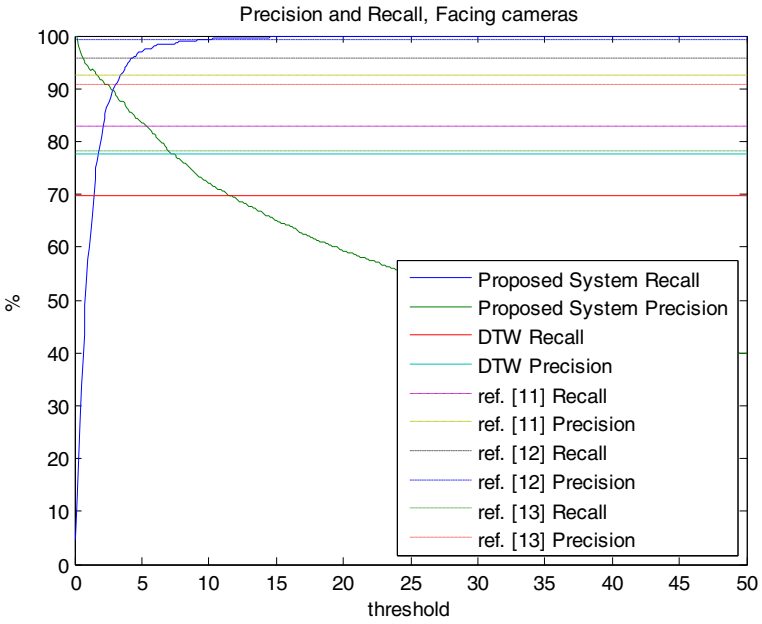


Fig. 13 Precision and Recall from fusion of facing cameras

between 5.4 and 6.1. Using the mcD&T module tracking, we get a value of 89.6 % for Precision and Recall, using a threshold value of 2.8. The other systems results for the

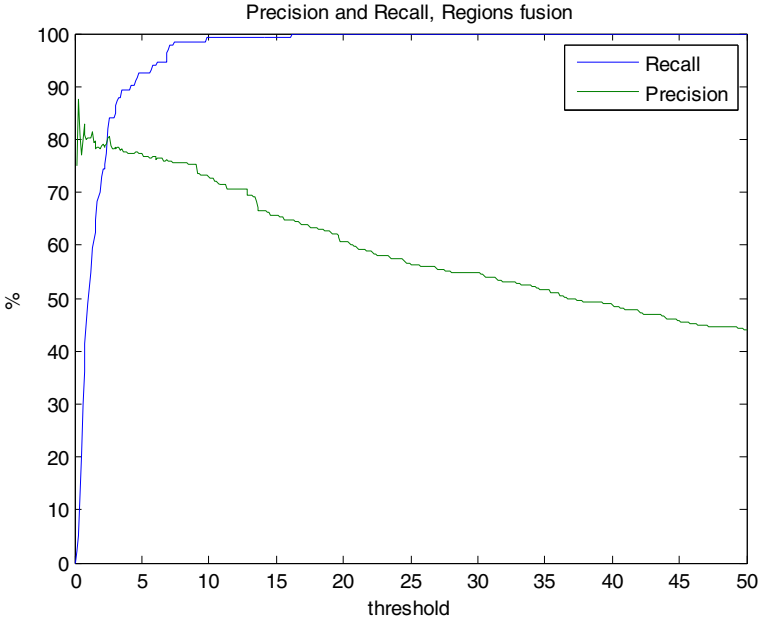


Fig. 14 Precision and Recall from fusion of the different resulting regions

Table 4 Time sequence time analysis

		Time (sec.)					
		Camera 1	Camera 2	Camera 3	Camera 4	Camera 5	Camera 6
Video reading and mcD&T		734	736	735	747	729	742
Data reading and Homographies		1	1	1	2	1	1
Fusion	Facing cameras	17		45		18	
	Regions	72					

comparison have been extracted from [1]. The DTW results are worse than our system results in both cases.

In the case of the gtD&T results, we get the best Precision and Recall values (100 %).

For the mcD&T module tracking results, in the equal error rate point (89.6 % Precision and Recall), we get better Recall result than [13] and [21], but a worse result for the Precision. For the Precision value of [13] (92.7 %), we get a Recall value of 76.2 % with a threshold value of 1.7, instead of the 83 % obtained by [13]. For the Recall value of [13] (83 %), we get a Precision value of 91.3 % with a threshold value of 2.1, just 1,4 % under the 92.7 % obtained by [13]. For the Precision value of [21] (90.7 %), we get a Recall value of 86.9 % with a threshold value of 2.5, instead of the 83 % obtained by [21]. For the Recall value of [21] (78.3 %) we get a Precision value of 92.1 % with a threshold value of 1.8, instead of the 90.7 % obtained by [21]. Note that our equal error rate precision value is only 1.1 % under the precision result of [21]. The presented results from [1] are better than our system results. The reference precision obtained by [1] is not reached in our system for a functional threshold. Table 5 summarizes the comparative values.

The objective of the work presented in this paper is to develop a simple starter system, based on semi-supervision, as most of the real systems, in order to overcome its current shortcomings, on which future improvements would be incorporated. Relatively good results are presented in the experiments, but the best systems results are not achieved. Using only position, we get better results than [21] and are close to [13], which use more complex parameters as polynomial regression trajectory models, graphs or Bayesian theory. The system presented in [1] shows better results than ours. As in the other systems, [1] uses more complex parameters than our system, as sharpness of turns or statistical trajectory characteristics. The advantage is that our system is simpler than the three systems used for the cross-validation [1, 13, 21]. The

Table 5 Precision and Recall result comparison

	Precision	Recall	Our Precision for the reference Recall	Our Recall for the reference Precision
[13]	92.7 %	83 %	91.3 % (−1.4 %)	76.2 % (−6.8 %)
[1]	99.3 %	95.6 %	71.7 % (−27.6 %)	Unreached reference precision
[21]	90.7 %	78.3 %	92.1 % (+1.4 %)	86.9 % (+8.6 %)

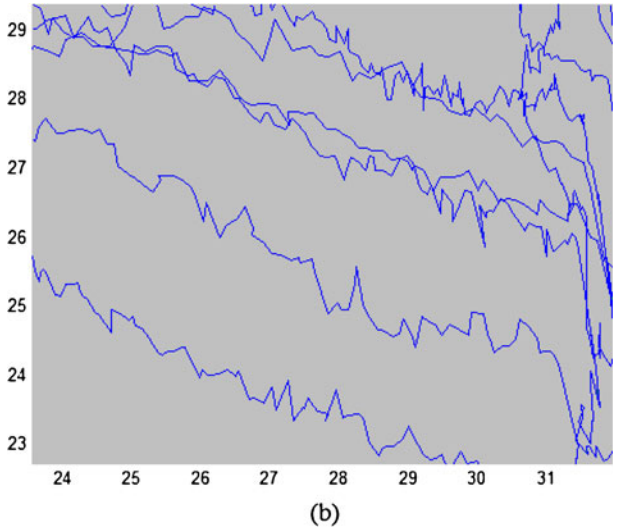


Fig. 15 Example of the zigzag effect in the trajectories

fusion process can be improved with multiple enhancements that would improve the precision and recall results.

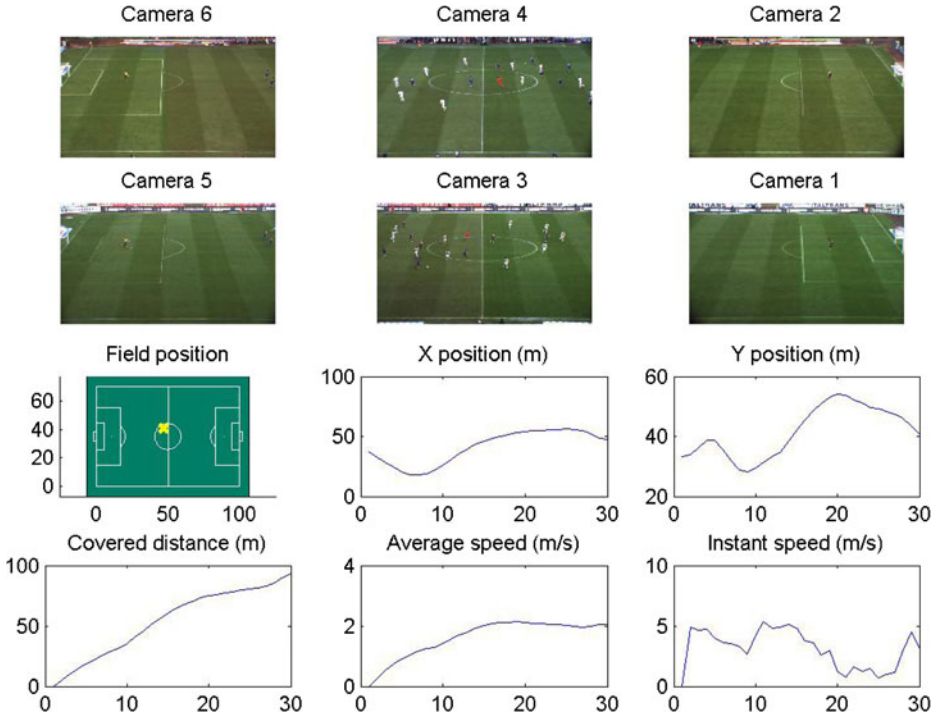


Fig. 16 Example of the resulting statistics video

6 Applications[8]

After obtaining the results of the system, some additional functionalities have been implemented to show some players statistics.

A zigzag effect occurs between two consecutive frames, as shown in Fig. 15. There are multiple sources of error that cause this problem: camera lenses, homographies, annotation, segmentation, tracking or fusion. This effect causes an error in the obtained statistics because they are higher than the real statistics. To reduce this error, the statistics are calculated every 25 frames, which corresponds to 1 s of video.

The statistics presented are: position on the X axis, position on the Y axis, covered distance, average speed and instant speed. The position on the X axis and the position on the Y axis are directly the coordinates of the player in a frame. The covered distance is calculated by adding the covered distance in each temporal interval (a temporal interval of 1 s, as indicated previously). The average speed is calculated by dividing the distance covered by the elapsed time. The instant speed is calculated as the average speed, but only considering the last temporal interval.

An example of the resulting statistics video is shown in Fig. 16 (the referee tracking is used to facilitate the visual tracking in the video). This video is available in the web page presented at the beginning of this section.

7 Conclusions and future work

7.1 Conclusions

The main objective of the work presented in this paper, the design and development of a system for detecting and tracking players in a field using multi-camera video, has been reached. After a previous configuration and with some supervision, the system is able to detect and track each player in the field, and to provide some statistics. The system is simple, complete, general and modular, easing future work improvements and modifications.

Sport videos with fixed cameras have significant characteristics that provide advantages for analysing them with respect to general videos of video surveillance (the mono-camera detection and tracking system was originally designed for that application domain). Backgrounds are generally static and uniform, except for certain areas such as public or dynamic advertising, which can be modelled relatively easily. People tracked have specific and distinct uniform, at least between the different types of people (players from each team, goalkeepers, referees …). This last case has a disadvantage, because the players on a team have exactly the same appearance as they wear the same uniform, which may complicate the tracking when occlusions occur.

The location of each camera is important. The ideal case is when the cameras are faced or symmetrically placed, since in these cases tracking errors are reduced in fusion process.

Better results are shown in cases with greater overlapping. This feature is observed in the case of team sports, when comparing the results of the facing cameras with the results from the fusion of regions.

Placing the cameras at higher elevations is interesting because it reduces the tracking error. The greater height of the camera, the smaller area of pixels is projected in the plane of the field and, therefore, ensures greater precision to the homography projection, but if the camera is located too high, the identification of the player uniform can be difficult.

[8] A web page has been created where some videos with the result of the system have been published: http://www-vpu.eps.uam.es/publications/ASemiSupervisedSystemForPlayersDetectionAndTrackingInMulticameraSoccerVideos/

In the case of team sports, the main problems are occlusions and regions with low or without overlapping. These systems are on which most improvement in needed, as seen in the state of art and in the experimental results. All the systems of this kind are prone to errors. The proposed system, thanks to its relative simplicity, can operate in real time computing partial trajectories.

There are many methods of fusion and many parameters which, when combined properly, may contribute to improve the results. As shown in the experiments, in the case of the real tracking system a supervised post-processing for obtaining the complete paths is required, whose replacement by an automatic process is proposed as future work.

7.2 Future work

Some future work lines are:

- *Background extraction*: The extractor used is relatively simple. Work could focus on making use of a more complex an extractor or one with lower computational cost, for example, selecting invariant pixels (or within margins) for a certain number of frames.
- *Graphical User Interface*: As described in the different sections, the initialization stage and the system analysis modules are configured manually making use of the corresponding scripts. Also for the supervised fusion of partial trajectories (see section 5.4), the tasks is currently done manually. A real-time interactive GUI should be developed for the system operators or system supervisors.
- *Tracking system*: This perhaps is the line with more possibilities for development. The system used was slightly adapted from one designed for video surveillance. This system is a good base, but there are many improvements and changes that could produce better results. Some proposals are:

  - *Camera configuration*: A study of the generalization to other camera layout configuration is proposed.
  - *Adaptive background generation*: Taking advantage of the specific characteristics of the background in each sport (public, dynamic advertising, …) the background can be updated during the macth.
  - *Detection of colours of the uniform of the players*: The values of Precision and Recall will increase using colour information in addition to the spatial information already used. For example, an additional constraint can be added to the fusion of two blobs, avoiding the fusion of pairs of blobs that should not be fused when corresponding to players of different teams.
  - *Real time implementation*: Optimizing and implementing the code in C++ will create a system that allows tracking players and fusing trajectories in real time.

- *Homographies:* The precision of each trajectory projection depends on the location in the field. An evaluation of the precision as a function of the position in the field and of the distance to the camera can prevent these precision differences, reducing the final error.
- *Fusion*: As mentioned previously, there are many parameters and methods for the fusion of the trajectories in team sports videos. Future work may consist of combining some existing methods or adding new ones. The post processing block to automatically connect fragmented resulting trajectories of the players in the team sports system is another line of future work.
- *Performance and Tactic analysis*: From the complete system, additional statistics and information can be calculated: areas of the field where the team stays longer, placement of players on the field, area of influence of each player, etc.

- *Statistics*: there are two main future work proposals for the statistics:
  - The statistics obtained by the system can be studied for evaluating the reliability, for example, using sensors in each player and comparing the results obtained by the sensors and the results obtained by the proposed system.
  - The zig-zag effect can be studied and corrected. A simple moving average window may be used to smooth the data.

Acknowledgments This work has been partially supported by the Spanish Government (TEC2011-25995).

References

1. Anjum N, Cavallaro A (2009) Trajectory association and fusion across partialy overlapping cameras. AVSS, pp 201–206
2. Bebie T, Bieri H (1998) SoccerMan-reconstructing soccer games from video sequences. Image processingl 1:898–902
3. Choi S, Seo Y, Kim H, Hong KS (1997) Where are the ball and players? Soccer game analysis with colorbased tracking and image mosaic. In: Proc. of ICIAP, pp 196–203
4. D'Orazio T, Leo M, Mosca N, Spagnolo P, Mazzeo PL (2009) A semi-automatic system for ground truth generation of soccer video sequences. AVSS, pp 559–564
5. de Meneses YL, Roduit P, Luisier F, Jacot J (2005) Trajectory analysis for sport and video surveillance. Electron Lett Comput Vis Image Anal 5(3):148–156
6. Du W, Hayet JB, Piater J, Verly J (2006) Collaborative multi-camera tracking of athletes in team sports. Workshop on Computer Vision Based Anal in Sports Environments (CVBASE), pp 2–13
7. Figueroa PJ, Leite NJ, Barros RML (2006) Tracking soccer players aiming their kinematical motion analysis. Trans Comput Vis Image Underst 101(2):122–135
8. Figueroa P, Leite N, Barros R, Cohen I, Medioni G (2004) Tracking soccer players using the graph representation. In: Proc. of ICPR 4:787–790
9. Hartley R, Zisserman A (2003) A multiple view geometry in computer vision. Cambridge University Press
10. Huang Y, Llach J, Bhagavathy S (2007) Players and ball detection in soccer videos based on color segmentation and shape analysis. Lecture Notes in Computer Science 4577:416–425
11. Junejo IN, Foroosh H (2007) Trajectory rectification and path modeling for video surveillance. In: Proc. of ICCV, pp 1–7
12. Kang J, Cohen I, Medioni G (2004) Tracking people in crowded scenes across multiple cameras. In: Proc. of ACCV
13. Kayumbi G, Anjum N, Cavallaro A (2008) Global trajectory reconstruction from distributed visual sensors. In: Proc. of ICDSC, pp 1–8
14. Kayumbi G, Mazzeo PL, Spagnolo P, Taj M, Cavallaro A (2008) Distributed visual sensing for virtual top-view trajectory generation in football videos. In: Proc. of CIVR
15. Martín R, Martínez JM (2013) An automatic system for sports analytics in multi-camera tennis videos. In: Proc. of AMMDS-AVSS (in press)
16. Misu T, Gohshi S, Izumi Y, Fujita Y, Naemura M (2004) Robust tracking of athletes using multiple features of multiple views. In: Proc. of WSCG, pp 285–292
17. Nummiaro K, Koller-Meier E, Svoboda T, Roth D, Van Gool J-L (2003) Color-based object tracking in multi-camera environments. In: Proc. of DAGM, pp 591–599
18. Poppe C, Bruyne SD, Verstockt S, de Walle RV (2010) Multi-camera analysis of soccer sequences. In: Proc. of AVSS, pp 26–31
19. Sachiko I, Hideo S (2004) Parallel tracking of all soccer players by integrating detected positions in multiple view images. In: Proc. of ICPR
20. SanMiguel JC, Martínez JM (2012) A semantic-based probabilistic approach for real-time video event recognition. Comp Vis Image Underst 116(9):937–952
21. Sheikh YA, Shah M (2008) Trajectory association across multiple airborne cameras. Trans Pattern Anal Mach Intell 30(2):361–367
22. Taj M, Cavallaro A (2009) Multi-camera track-before-detect. In: Proc. of ICDSC
23. Tong X, et al (2004) An effective and fast soccer ball detection and tracking method. In: Proc of ICPR 4:795–798
24. Xinguo Y, Farin D (2005) Current and emerging topics in sports video processing. In: Proc. of ICME
25. Xu M, Orwell J, Jones G (2004) Tracking football players with multiple cameras. In: Proc. of ICIP 5:2909–2912

Rafael Martín Nieto was born in Madrid, Spain, in 1989. He studies Ingeniero de Telecomunicación degree (5 years engineering program, from 2007 to 2012). He received excellent academic performance to degree studies (given by Comunidad de Madrid) in the years 2007, 2008 and 2010. Besides he worked as collaborator at the VPULab from 2007 until 2011.

José M. Martínez received the Ingeniero de Telecomunicación degree (6 years engineering program) in 1991 and the Doctor Ingeniero de Telecomunicación degree (PhD in Communications) in 1998, both from the E.T.S. Ingenieros de Telecomunicación of the Universidad Politécnica de Madrid. He is Associate Professor at the Escuela Politécnica Superior of the Universidad Autónoma de Madrid. His professional interests cover different aspects of advanced video surveillance systems and multimedia information systems. Besides his participation in several Spanish national projects (both with public and private funding), he has been actively involved in European projects dealing with multimedia information systems applied to the cultural heritage (e.g., RACE 1078 EMN, European Museum Network; RACE 2043 RAMA, Remote Access to Museums Archives; ICT-PSP-FP7-250527 ASSETS, Advanced Search Services and Enhanced Technological Solutions for the Europeana Digital Library), education (e.g., ET 1024 TRENDS, Training Educators Through Networks and Distributed Systems), multimedia archives (e.g., ACTS 361 HYPERMEDIA, Continuous Audiovisual Digital Market in Europe) and semantic multimedia networked systems (e.g., IST FP6-001765 acemedia, IST FP6-027685 Mesh). He is author and co-author of more than 100 papers in international journals and conferences, and co-author of the first book about the MPEG-7 Standard published 2002.

He has acted as auditor and reviewer for the EC for projects of the frameworks program for research in Information Society and Technology (IST). He has acted as reviewer for journals and conferences, and has been Technical Co-chair of the International Workshop VLBV'03, Special Sessions Chair of the International Conference SAMT 2006, Special Sessions Chair of the 9th International Workshop on Image Analysis for Multimedia Interactive Services WIAMIS 2008, Program co-chair of the 7th International Workshop on Content-based Multimedia Indexing CBMI 2009 and General chair of the 9th International Workshop on Content-based Multimedia Indexing CBMI 2011 (he also co-edited the associated Special Issue in Multimedia Tools and Applications).