Journal of Real-Time Image Processing manuscript No.
(will be inserted by the editor)

Consensus-based trajectory estimation for ball detection in calibrated cameras systems

Pascaline Parisot · Christophe De Vleeschouwer

Received: date / Accepted: date

Abstract This paper considers the detection of the ball in team sport scenes observed with still or motion-compensated calibrated cameras. Foreground masks do provide primary cues to identify circular moving objects in the scene, but are shown to be too noisy to achieve reliable detections of weakly contrasted balls, especially when a single viewpoint is available, as often desired for reduced deployment cost. In those cases, trajectory analysis has been shown to provide valuable complementary information to differentiate true and false positives among the candidates detected by the foreground mask(s). In this paper, we focus on the detection of ball trajectory segments, exclusively from visual cues, without considering semantic reasoning about team-play to connect those segments into long trajectories. We revisit several recent works, mainly the ones that we presented in [1] and [2], and introduce a publicly available dataset to compare them. We conclude that randomized consensus-based methods are competitive compared to the alternative deterministic graph-based solutions, while offering the additional advantage to naturally extend to the cost-effective single view scenario. As an original contribution, we also introduce a procedure to efficiently clean up the foreground mask in correlation-based methods like [1], and a non-linear rank-order filter to merge the foreground cues from multiple viewpoints. We also derive recommendations regarding the camera positioning and the buffering needs of a real-time acquisition system.

Keywords ball detection · RANSAC · single view

P. Parisot and C. De Vleeschouwer
ICTEAM-ELEN,
Université catholique de Louvain,
Place du Levant 2,
1348 Louvain-La-Neuve, Belgium
E-mail: christophe.devleeschouwer@uclouvain.be

1 Introduction

Our work investigates how the ball can be detected in team sports actions, in a reliable but cost-effective manner, using a minimum number of cameras distributed around the sport field. Whilst our contributions are demonstrated in a static acquisition set-up, they only assume that the objects moving in the scene can be extracted with reasonable reliability, using off-the-shelf background subtraction algorithms. Hence, the investigated methods remain valid for motion-compensated pan/tilt/zoom camera views. From an application perspective, locating the ball (and the players [3,4]) in a sport context appears to be relevant either to feed sport analytics and enrich broadcasted content [5], or to drive automatic team sport coverage [6–8].

Beyond the sport analysis context, the detection of objects-of-interest is a central component of most video analysis systems dedicated to scene interpretation. Simplest detectors traditionally build on the extraction of connected components in a foreground mask [9–12], and on the comparison of their shapes with a model of the object silhouette [13,14]. Those detectors obviously suffer from significant false and/or missed detection rates in presence of multiple moving objects, or when the object of interest is not moving. To address those limitations, modern approaches have proposed (i) to merge the information carried by the foreground masks computed from multiple and complementary views of the same scene [15–17,4], or (ii) to exploit visual classifiers to capture the specificities of the object visual appearance in terms of texture patterns [18–20].

In practice however, those modern approaches require either the deployment of a multi-view set-up, or the identification of visual features that are able to discriminate the object-of-interest from the rest of the scene. The first option dramatically increases the deployment cost since, by multiplying the number of cameras, it increases the amount of processed data, requires calibration, and makes the system connectivity more complex. The second option is not more practical since it relies on a representative set of training samples to learn the object appearance, which inherently restricts the scope and the genericity of the resulting detector, and induces a cost to collect and annotate a training set that is relevant to the application at hand [3]. More importantly, in case of ball detection, the discriminative power of visual features is generally tightly coupled to the contrast between the scene and the ball. Since reliable ball detection solutions have already reached the market in highly-contrasted scenarios (tennis and soccer [21–23]), in this paper we primarily investigate solutions that provide reliable detection of weakly contrasted balls in a cluttered scene observed from a limited number of viewpoints. To reduce the implementation cost and the overall computational complexity, we are especially interested in solutions that can live in single viewpoint acquisition set-ups, using exclusively background-subtracted visual cues.

As main contributions, our paper presents:

- an original strategy to process the foreground mask(s) from a (set of) static view(s) to infer the presence of a small moving object in the image(s) captured at a given time instant. The approach extends our initial work in [1]. As in [1], it builds on integral images for improved computational efficiency. Beyond [1] and to improve the detection accuracy, it introduces an original foreground mask(s) players-aware clean-up mechanism to isolate ball candidates, and a novel non-linear filter to merge the ball candidate position information from multiple viewpoints. Also in addition to [1], it recommends the use of spherical coordinates to approximate the ball silhouette template. When defined in terms of angular aperture, the ball silhouette template indeed remains circular whatever the position of the ball in the scene. In contrast, when described in terms of image plane coordinates, the projected ball silhouette becomes elliptical when the ball moves away from the principal axis of the camera. Eventually, and still in complement to [1], our paper also demonstrates the real-time implementation feasibility of our solution, and reveals that a minimal delay of about half a second is desirable to achieve accurate detection in a real-time acquisition scenario;
- a comparative analysis of graph- and consensus-based approaches to estimate a ballistic trajectory from the corrupted set of ball candidates obtained along the time. Such comparison was missing in earlier literature. To fill this gap, we release a novel and publicly available groundtruth for the APIDIS dataset [24], and builds on it to compare a variety of use cases. Our investigation reveals that the stochastic consensus-based regression has the advantage to naturally generalize to the single-view scenario, whilst reaching similar performances than the exhaustive graph-based search in the multi-view scenario.

When combined together, all those contributions allow for cost-effective and reliable detection of the ball trajectory segments, starting from a computationally efficient foreground mask analysis, and increasing the detection reliability based on advanced processing of a limited number of candidates. Our paper primarily focuses on the practical deployment aspects. Since our method is based on visual cues that are extracted from the part of the foreground mask that are disjoint from the player silhouettes, it can detect the ball only when the ball is not occluded or held by a player. Our method can however obviously be augmented with any kind of additional high-level semantic reasoning about team play, to connect the detected tracklets into long term ball trajectories [25–27].

The rest of the paper is organized as follows. Section 2 positions the previous art related to ball detection with respect to our challenging applicative context (low contrast, scene cluttered with many team sport players). In particular, it motivates the relevance of the comparative study provided in our experimental validation. Section 3 explains how the specific shape and the high speed of the ball support the formal definition of a ball detector. Primarily, it extends our work in [1], by introducing a players-aware pre-processing of the foreground mask to locate the rapidly moving objects at individual time instants. It also proposes a non-linear strategy to fusion the foreground cues in the multi-view case. Section 4 analyzes the trajectory of the ball candidates to identify the ones following a ballistic trajectory. It generalizes the consensus-based ballistic trajectory estimation to the single-view case, and compares it to graph-based approaches in the multi-view case. Section 5 considers a representative use case, and provides a comparative analysis of relevant state-of-the-art methods. The study validates our contributions, and demonstrates that the proposed consensus-based approach favorably compares to graph-based approaches in the multi-view case, while naturally generalizing to the single-view case. The experimental results also provide the required material to guide some of the critical design choices (i.e. number of views, buffer requirements, camera positioning with respect to the scene) involved in the deployment of a static video system for team sport analysis. Section 6 concludes.

2 Related works

This section presents the previous works that are relevant to our study, and positions them with respect to our envisioned application scenario, both in terms of detection effectiveness and deployment constraints.

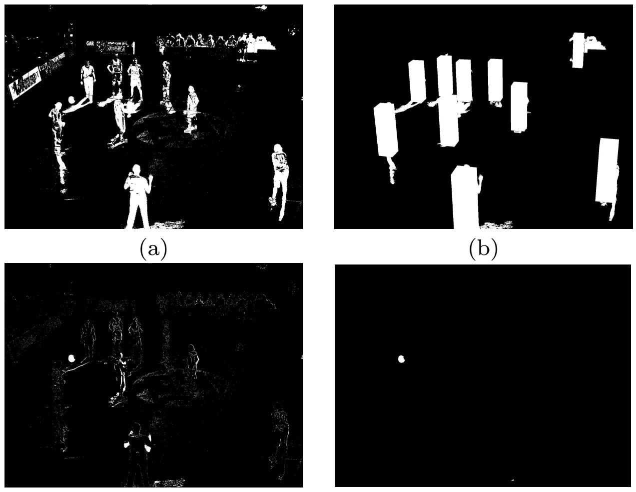
The ball is of particular interest in many sports, and its detection has been largely investigated, leading to industrial products like the automated line calling system in tennis or the goal line technology in soccer [21]. In tennis, table tennis, or soccer games, the ball is visible most of the time, which makes the track-before-detect approaches relevant [22,23,28,29]. The same holds when the system aims at detecting whether a ball crosses a field border [30]. In contrast, in sports like basketball, the ball is often occluded by players, and a detect-before-track strategy is generally adopted [31,1,2]. We focus on those solutions in the rest of the paper.

Most of the methods proposed in the literature exploit the circularity and ballistic trajectory of the ball.

Primarily, the ball is envisioned as a small circular and moving object, potentially characterized by a specific color pattern. Hence, it is generally segmented in individual images based on the identification of circular shapes in a foreground mask [31–34]. For improved accuracy, but at the cost of additional computational complexity, the authors in [26] scan all the foreground pixels in a sliding-window manner, obtain the color-histogram of the current window and compute the inner-product between a reference ball template histogram and the current histogram. The actual detection performance significantly depends on the contrast between the ball and the background, which might be high (volley, tennis or soccer) or low (basket-ball). We will focus on the low-contrast case in this paper, since any detection algorithm working on an individual image in this case induces a significant rate of false/missed detections [1,2], including when multiple views are available.

To increase the detection reliability for low-contrasted balls, most previous works analyze the displacement of a conservative set of ball candidates across time, to only retain the candidates that follow a ballistic trajectory. Specifically, Chen et al. [31] consider the shooting actions of a basket-ball game, as displayed in highly contrasted TV broadcasted content. They analyze the trajectory of the candidates by fitting consecutive detections to a parabolic model, solving a quadratic minimization problem. In [34], the same authors connect closely detected volley-ball candidates along the time, and only retain the sequence of detections that fit a parabolic equation. [35] adopts the same strategy to detect long shots in basket-ball sequences. Our previous work in [1] follows a similar idea, but adopts a RANSAC approach to deal with the lack of accuracy and reliability of the initial candidate detections obtained in cluttered scenes, when the ball lies among the players and is weakly contrasted compared to the background. As an alternative to the quadratic regression, graph-based formalisms have also been considered to formulate the ball trajectory estimation problem. In Yan et al. [32], for highly contrasted balls, a set of ballistic trajectories are first estimated based on triplets of consecutive candidates, and a graph-based formalism is adopted to infer a global trajectory from those local triplets trajectories. Since it relies on triplets of consecutive detections, the method is not appropriate when only few detections are available along the time, which is the case with low-contrasted balls. Zhou and al. [33] follow the work in [32] in that they also attempt to connect sub-trajectories into global ones. They also define sub-trajectories based on the projection of local motion models, which requires high detection rates on individual images, as only encountered in highly contrasted context. To circumvent this limitation, the work we have initially presented in [2] has proposed to estimate the ball trajectories by building a so-called velocity graph. In this graph, each node represents the 3D velocity measured between a pair of temporally ordered 3D candidate detections. Each edge compares the acceleration measured between the two nodes with the gravity acceleration, so that short paths in the graph correspond to ballistic trajectories. The approach appears to be both elegant and effective, but relies on 3D positions as input ball candidates. Hence, it cannot be used in the case of a single view acquisition set-up.

Players positions provide indirect and complementary cues to locate the ball in many kinds of games. [36] and [37] have shown that the location of the ball can reasonably be predicted based on players displacement analysis, but do not merge this information with cues related to direct detection of the ball. Explicit heuristic rules have also been considered to infer which player holds the ball from independent tracking of the ball and the players [38,39]. The most relevant works with respect to ball tracking are probably the ones that have considered the joint tracking of players and ball [25–27, 40]. Those works reveal that, by estimating jointly and globally the trajectories of interacting players and ball, the presence of the ones (most often the ball) that are not initially detected based solely on image evidence can be inferred from the detections of the others. In particular, in their recent and innovative works, Wang et al. have formulated explicitly the fact that the ball can appear or disappear at players locations. The work in [26] uses Conditional Random Fields (CRF) to estimate whether the ball lies between players or is in possession of one of those players. It takes as input (i) the position of the players along the time, and (i) a ball occcupancy map (BOM). In practice, the BOM is derived from a multi-view ball detection tool that is similar to one of the algorithms investigated and compared in our paper. Hence, the lessons drawn from our work will directly benefit the application of the framework presented in [26], including in single view scenarios. The same remark holds for [27,40], which formulate the joint players/ball tracking problem as an integer program. In this formulation, flow conservation constraints ensure that the ball appears from the same player where it disappeared. Those two frameworks are strongly recommended to leverage any visual ball detector to achieve long term tracking. They thus effectively complement our approach to handle long-term tracking when the ball is kept in players' hand. They however both directly depend on the capacity to derive reliable cues about the presence and localization of the ball in the image(s). This makes especially relevant our careful investigation and comparison of the different approaches proposed in the literature to estimate ball trajectory segments from visual cues.



(a) (b) (c) (d)

Fig. 1 Masks obtained for Camera 7, APIDIS dataset: (a) the foreground mask F_t^c; (b) the players mask M_t^c; (c) the mask $\overline{F_t^c}$ defined in Eq. (1); (d) the players-aware cleaned-up foreground masks $\widetilde{F_t^c}$, which only keeps the connected components in (c) that are disconnected from the ones in (b).

3 Lowly-contrasted ball candidates detection

This section extends our previous works in [1] to detect the ball independently in each frame, in the challenging case for which the contrast between the ball and the background is too small to achieve a reliable detection from a simple (motion and) color thresholding. Without loss of generality, we present our approach in a multi-view set-up, and highlight the single-view case specificities when relevant.
The rest of the section is organized in two parts:

Section 3.1 assumes that players location are known[1] and explains how the foreground mask derived from a conventional background subtraction algorithm can be processed to only retain the silhouettes that move rapidly between the players.

Section 3.2 computes a set of ball position candidates at each time instant, based on the analysis of the processed mask.

3.1 Players-aware extraction of rapidly moving silhouettes

This section presents how the foreground mask derived from a conventional background subtraction algorithm can be processed to only retain the foreground silhouettes that likely correspond to the ball. It considers the ball to be a small, rapidly moving, circular object, and assumes that the players' locations are known. This assumption is realistic since [4], optionally combined with [3] in a single-view acquisition set-up, achieves -in real-time on a conventional PC- detection rates higher than 95%, with less than a few percent of false positives.

Figure 1 illustrates our mask simplification process. Input foreground masks are computed in each view using the background subtraction algorithm in [9], with a relatively small foreground detection threshold, so as to detect the ball even in case of a weakly contrasted scene[2]. Given the foreground masks F_t^c and F_{t-1}^c computed in camera c at times t and $t-1$, we first compare F_t^c and F_{t-1}^c to only retain the foreground pixels that are present in the first mask, but not in the second one. Formally, we define the masks difference $\overline{F_t^c}$ as:

$$\overline{F_t^c} = F_t^c \wedge \neg F_{t-1}^c \tag{1}$$

with $\wedge$ and $\neg$ denoting the 'AND' and 'NOT' operators. As observed in Figure 1(c), this masks only retains the foreground pixels from F_t^c that are uncovered, *i.e.*, part of the background, at time $t-1$, thereby only highlighting objects whose projection in view c moves rapidly.

[1] In this paper, the player positions are computed as in [4].

[2] The use of a small threshold induces a significant amount of false positive foreground pixels in the mask, which makes the clean-up stage, as well as the subsequent template-based correlation and ballistic trajectory analysis, especially helpful to avoid false ball detections.

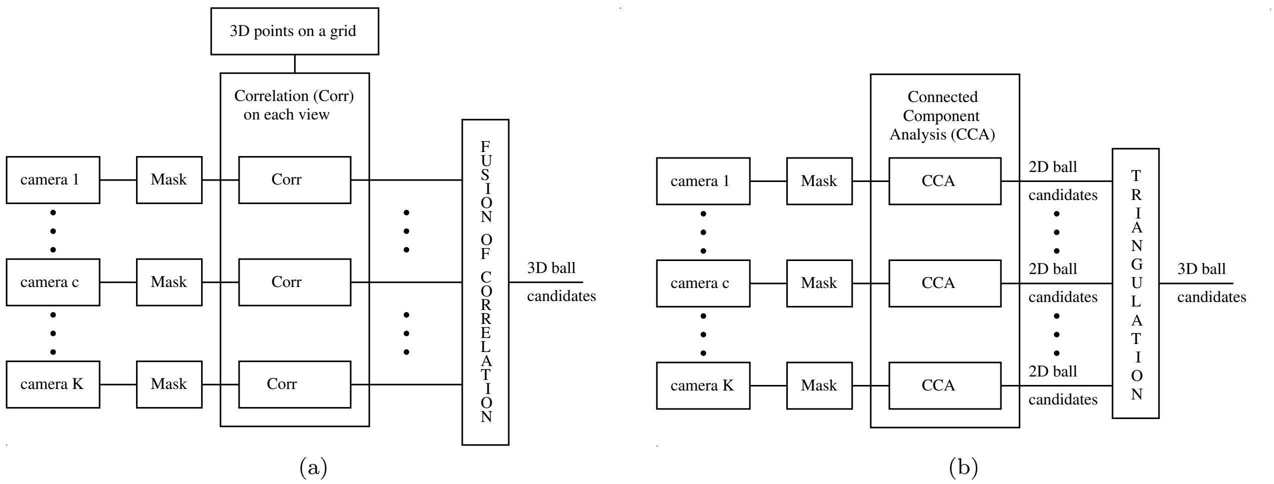


Fig. 2 Two frameworks for multi-view ball detection: (a) Corr: the correlations of each of the 2D foreground masks with an approximated template of the ball silhouette are merged to identify the 3D locations that are likely to support the ball; (b) CCA: the shape and size analysis of the connected components selects small circular objects in the 2D foreground masks, which are then triangulated to infer 3D ball position candidates.

To refine $\overline{F_t^c}$, the moving foreground pixels that are part of the players silhouettes are removed. This is done by considering the players mask M_t^c, which defines, for the view c, the areas of the image covered by players at time t. Knowing the positions of the players on the ground plane [4], the players mask is computed as the union of the projections on the view of the 3D players silhouettes, as approximated by 3D bounding boxes[3]. The connected components of the foreground mask F_t^c that are linked to those projected boxes are mainly due to players shadow and reflection, and are thus added to the players mask[4]. The players-aware cleaned-up foreground mask $\widetilde{F_t^c}$ is then simply computed by subtracting M_t^c from $\overline{F_t^c}$.

3.2 Instantaneous detection of ball position candidates

Figure 2 presents two approaches to compute a set of ball position candidates at a given time instant t, based on the analysis of the players-aware cleaned-up masks $\widetilde{F_t^c}$. The first method extends the correlation-based approach we initially introduced in [1], while the second one follows the connected component analysis framework, introduced in our work [2], and similar to the approach adopted in [26]. In both approaches, we only consider the foreground mask and ignore the color cue. This is because (i) color appears to be poorly discriminant in lowly contrasted indoor basket-ball scenes, and (ii) working with the foreground binary mask reduces the ball template matching computation to a few operations when using integral images, which is especially attractive regarding real-time implementation.

3.2.1 Correlation-based detection method

This method considers integral images to efficiently compute the ball position candidates from the correlation between the mask $\widetilde{F_t^c}$ and an approximated template of the expected circular silhouette of the ball.

Fundamentally, our proposed method considers a 3D grid of possible 3D positions for the ball center, and estimates the likelihood that the ball is actually located in one of these positions. We first explain how the likelihood is estimated in one single view, and then present our non-linear approach to merge the likelihood from multiple views.

Given a calibrated camera c, the projection of a ball centered in $M \in \mathbb{R}^3$ corresponds to a disk with center $m_c \in \mathbb{R}^2$ and radius $r_{c,M}$, where $r_{c,M}$ can be computed based on the calibration parameters of the camera c, the center M of the ball and the known ball size. Thus, the likelihood that a ball is located in M can simply be estimated by correlating the mask $\widetilde{F_t^c}$ with the template $T^{(c,M)}$ (see Figure 3-(a)) which is a white disk, with center m_c and radius $r_{c,M}$, surrounded by a black annulus, whose radius has been set to $2r_{c,M}$. To give the same weight to the inner and outer disks, their constant values are set to 1 and 1/3, respectively.

[3] Each player is approximated by a 40 cm (width) × 180 cm (height) rectangular box.

[4] To prevent the unfounded removal of ball candidates, the operating point of the player detector is chosen to maintain a very small false positive rate, at the cost of a relatively moderate detection rate.

In practice, to guarantee that the ball silhouette remains circular when the ball moves away from the principal axis of the camera, we recommend to work in spherical coordinates (centered in the camera center), instead of image plane coordinates as was done in our initial work [1]. For increased computational efficiency, the circular template is approximated by a squared one (see Figure 3-(b)), so that correlations can be computed based on (spherical) integral images [41]. The conventional integral image algorithm is used, but is implemented on images defined in terms of the polar and azimuthal coordinates of the spherical coordinate system instead of the conventional x and y image plane coordinates. For camera c, the (angular aperture) size $r_{c,M}$ of the template associated to the position M is set to $\arcsin(b/d_c) \approx b/d_c$, where d_c denotes the distance between M and the center of camera c, while b denotes the radius of the ball. The size of the external side of the back area is set to $\sqrt{2}\ .\ r_{c,M}$, so that the black area equals the white area. To compensate for the variation of the template size with the distance between the position M and the camera, those correlations are divided by a normalization factor. We chose it equal to twice the inner square area, so that the normalized correlation lies in $[0,1]$. We let $C[c](M)$ denote the normalized correlation in camera c, at position M.

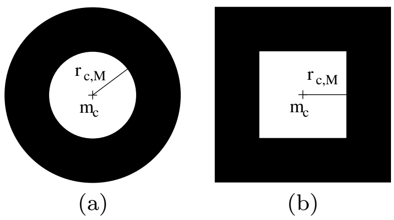


Fig. 3 Templates to compute the correlation measure: (a) Exact template $T^{(c,M)}$; (b) Approximated template $\widetilde{T^{(c,M)}}$ to use integral images.

We now explain how the normalized correlation metrics from multiple views are combined to infer a joint likelihood of ball presence in M. We consider two strategies.

The first one follows the linear approach introduced in [1]. It simply sums up the normalized correlations computed in each view for position M, and defines the joint correlation $C_{tot}(M)$ to be:

$$C_{tot}(M) = \frac{\sum_{c \in \mathcal{K}} C[c](M)}{|\mathcal{K}|} \tag{2}$$

where $\mathcal{K}$ is the set of camera index for which m_c lies inside the field of view, without being hidden by a player, and $|\mathcal{K}|$ is the cardinal of the set $\mathcal{K}$. A position M is defined to be a ball position candidate as long as $C_{tot}(M) > \tau_{tot}$.

The second approach is original, and proposes a non-linear rank ordering approach to fusion the normalized correlations. This is to account for the fact that the low-contrasted nature of the ball (which induces missed detections), together with the calibration inaccuracies and the potentially loose synchronization between the cameras (as encountered with IP cameras for example), makes it very unlikely to observe consistent ball silhouettes in every view. Therefore, we propose to relax the ball presence likelihood estimator by considering that the ball is likely to be present in M as long as the correlations $C[c](M)$ are above some threshold τ_{corr} in at least $r \leq |\mathcal{K}|$ views. Equivalently, this happens when the r^{th} rank order correlation $C_{RO[r]}(M)$ lies above τ_{corr}.

In summary, the correlation-based detection approach builds on our previous work in [1], but extends it by lie in the fact that:

- [1] estimates the ball position based on the foreground mask F_t^c, and not based on a players-aware cleaned-up mask $\widetilde{F_t^c}$;
- introducing a non-linear rank-order filter to fusion the information computed in different views, which is shown to advantageously replace the linear filter presented in [1] when the ball silhouette is likely to be occluded in the multiple views available.

3.2.2 Detection based on connected component analysis

We now briefly survey the approach we have introduced in [2]. The method simply selects the regions that are reasonably circular, and have a size that is close to the expected ball size in camera c. Hence, it removes from $\widetilde{F_t^c}$ the connected components that are bigger than τ_{big} pixels or smaller than τ_{small} pixels, and the ones with an aspect ratio that is higher than τ_λ. The thresholds are determined based on the camera calibration parameters, relaxed based on the statistical analysis of a few ball samples. In a multi-view set-up, where the game is observed by several (loosely) synchronized and calibrated cameras, we fusion the partial decisions taken in each view by triangulating the pairs of 2D candidates extracted in different views to define 3D candidates. Specifically, given the 2D ball candidate locations extracted in each view, the triangulation works in two steps. In the first step, all pairs of 2D ball candidates extracted in two different views are triangulated to define a set of 3D triangulated positions. When the light rays associated to the 2D candidates do not intersect, the 3D point that lies in the middle of the shortest segment connecting the two rays is selected as a 3D triangulated position only if its distance to the light rays is smaller than 30 cm. In a second step, the 3D triangulated positions that are sufficiently close to each others (less than τ_{close} centimeters) are merged. In practice, this is done using a simple greedy approach that first selects the 3D point with the largest number of sufficiently close neighbors, and then iterates on the residual set of 3D triangulated positions.

3.2.3 Complexity analysis, and single view case simplifications

The correlation-based (Corr) and the connected component analysis (CCA) methods fundamentally differ in the way they process the foreground masks. The correlation approach scans a relatively dense grid of 3D points, and correlates each one of the 2D views with an approximated template of the ball silhouette. Such a process is computationally realistic in a limited 3D area, e.g. around the basket or the goal in a basket-ball or soccer game for example, but becomes intractable over large scenes, since the number of correlations to compute increases proportionally to the volume of the investigated area.

In contrast, the CCA approach starts by picking-up 2D ball position candidates from 2D images, and then triangulates to infer 3D candidates. This method might suffer when the ball silhouette is split into several pieces of foreground mask, but it scales reasonably well with the size of the scene since the complexity associated to the detection of 2D candidates is proportional to the size of the foreground mask, and not to the volume of the scene. This makes it worth evaluating the penalty induced by this method in terms of detection accuracy, compared to a correlation based approach. This analysis is carried out in Section 5.1.

Eventually, particular case of interest is the one of a single panoramic side view. This case is widespread among commercial automatic sport production and analysis solutions, e.g. [8] for basket-ball production or [42] for soccer analysis. In this case, the ball silhouette has a relatively constant size, because the camera is far from the 3D regions in which the ball has to be detected[5]. As a consequence, the correlation-based approach can be implemented efficiently as a simple convolution, which mitigates the complexity drawback associated to the Corr method.

[5] Assuming that the single viewpoint is at least 15m away from the field, which is common in practice, the actual template size lies within a margin of less than 30% around the constant-size template, chosen to fit the ball silhouette when the ball is in the field center.

4 Inferring ballistic trajectories from ball position candidates

To identify false positives among the candidates extracted in the above section, as well as to interpolate the missed detections, this section explains how to infer plausible ballistic trajectories from the ball candidate positions.

As detailed in Section 2, both least-squares function approximation and graph-based approaches have been considered in the past for this purpose. In this paper, we aim at studying and comparing those two strategies in terms of tracking effectiveness, but also deployment constraint. Therefore, we review and extend the state-of-the-art methods we have respectively introduced in [1] and [2] to address each one of those two strategies.

4.1 Consensus-based trajectory estimation

Our RANSAC-based approach estimates a ballistic trajectory from a consensus between the quadratic regressions computed from random subsets of ball position candidates in a temporal neighborhood. In terms of deployment, the method has the advantage to naturally generalize to the single view scenario.

4.1.1 Motion model and least squares approximation

Neglecting the friction forces, a ball is only subject to the gravity force, inducing the acceleration $\mathbf{a} = \mathbf{g} = [0 \;\; 0 \;\; -g_z]^T$ with $g_z = 9.81$ m/s^2. The ballistic trajectory model is given by:

$$\begin{cases} f : \mathbb{R} \to \mathbb{R}^3 \\ \quad t \mapsto f(t) = \frac{1}{2}\mathbf{g}t^2 + \mathbf{v}t + \mathbf{p} \end{cases} \tag{3}$$

where t is the time, and $f(t) = \begin{bmatrix} x(t) \; y(t) \; z(t) \end{bmatrix}^T$, $\mathbf{v} = \begin{bmatrix} v_x \; v_y \; v_z \end{bmatrix}^T$, $\mathbf{p} = \begin{bmatrix} p_x \; p_y \; p_z \end{bmatrix}^T$ ($\in \mathbb{R}^3$) are respectively the coordinates of the point at time t, the initial velocity and the initial position.

Given a set of N 3D points $\{(x_i, y_i, z_i)\}_{i=1..N}$ defining the ball center at time t_i, the 6 model parameters $\beta = \begin{bmatrix} v_x \; v_y \; v_z \; p_x \; p_y \; p_z \end{bmatrix}^T$ are estimated by solving the following linear least squares problem:

$$\hat{\beta} = \begin{cases} argmin \quad ||A\beta - b||^2 \\ \beta \in \mathbb{R}^6 \end{cases} \tag{4}$$

where $A \in \mathbb{R}^{3N \times 6}$ and $b \in \mathbb{R}^{3N}$ are defined as:

$$A = \begin{bmatrix} t_1 & 0 & 0 & 1 & 0 & 0 \\ 0 & t_1 & 0 & 0 & 1 & 0 \\ 0 & 0 & t_1 & 0 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ t_N & 0 & 0 & 1 & 0 & 0 \\ 0 & t_N & 0 & 0 & 1 & 0 \\ 0 & 0 & t_N & 0 & 0 & 1 \end{bmatrix}, b = \begin{bmatrix} x_1 \\ y_1 \\ z_1 + \frac{1}{2} g_z t_1^2 \\ \vdots \\ x_N \\ y_N \\ z_N + \frac{1}{2} g_z t_N^2 \end{bmatrix} \tag{5}$$

4.1.2 Single-view ballistic trajectory estimation specificities

In the single view case, the ball position candidates are defined in 2D image coordinates, so that the estimation of the six parameters of the ballistic trajectory is not straightforward anymore. However, our method easily generalizes to this case by taking the projection matrix of the calibrated camera into account, as follows. Let us consider a set of N 2D points $\{(u_i, v_i)\}_{i=1..N}$, each point defining the center of the ball in the single view at time t_i.

Given the projection matrix $P \in \mathbb{R}^{3\times4}$ of the camera and a point $M = \begin{bmatrix} x(t) & y(t) & z(t) \end{bmatrix}^T$, its projection $m = \begin{bmatrix} u(t) & v(t) \end{bmatrix}^T$ is given by:

$$\begin{bmatrix} \lambda u(t) \\ \lambda v(t) \\ \lambda \end{bmatrix} = P \begin{bmatrix} x(t) \\ y(t) \\ z(t) \\ 1 \end{bmatrix} \tag{6}$$

By replacing $x(t)$, $y(t)$ and $z(t)$ in Equation (6) based on $f(t)$ in Equation (3), we obtain a system of the form defined in Equation (4), with $A_{2D} \in \mathbb{R}^{2N\times6}$ and $b_{2D} \in \mathbb{R}^{2N}$.

4.1.3 Statistical-based ballistic trajectory estimation

A RANSAC (RANdom SAmple Consensus) approach is adopted to remove outliers among the detected ball candidate positions. The RANSAC process [43] is applied on a sequence of windows of size T, shifted by steps of size s, so that the ball position is estimated in s novel frames at each step. T is typically set to cover a one second period (T=24 in our experiments), while s is typically set to $T/2$. To improve the detection accuracy, we tolerate a delay of d frames, meaning that the ball position at time t has to be estimated at the latest when frame $t+d$ becomes available. In practice, since the window progresses by steps of size s, we have to estimate the ball position between frames $k.s$ and $(k+1).s$ as soon as the frame $k.s+d$ becomes available. Since we want to maximize the number of future frames that are available when estimating the ball position at any time t, s is set to one. Section 5.2 analyzes the impact of the delay parameter d. Each iteration of the RANSAC process first estimates the motion model parameters from $n_{ini} = 3$ ball candidate positions taken at random. To reduce the impact of the noise affecting the candidate positions[6], random candidates are selected such that they are at least separated by $n_{step} = 2$ timestamps. In a second step, the candidate positions that fit this estimated model (with a certain threshold $\tau_{fit} = 30$ centimeters) are considered as inliers; the model is validated and is re-estimated from these inliers if their number is larger than a threshold $n_{inliers}$, set to $max(4, l/2)$ with l being the length of the trajectory model, as defined by the largest inliers frame indices difference. Finally, the mean squared error between the estimated model and the inliers is computed. This procedure is repeated for a determined number of iterations $n_{ite} = 500$ and the model with minimal mean squared error is kept, if the error is smaller than a threshold $\tau_{cost,RANSAC}$ centimeters. This threshold fixes the operating point of the method in terms of trade-off between the detection rate and the false positive rate. Its variation allows to plot ROC curves in Section 5.2.

A number of additional criterions are considered to discard the trajectories that obviously do not correspond to the ball, either because their vertical amplitudes are flat (assessed based on appropriate thresholds) or because their occupancy rates are lower than 50%. The occupancy rate is the ratio between the number of 3D candidates close to the detected ballistic trajectory and its duration in timestamps. The same criterions are used to post-process the tracklets derived from the graph-based method presented below.

4.2 Graph-based trajectory estimation

This section considers a graph-based formalism to define ballistic tracklets from the sequence of 3D ball candidates. Graph formulations have been considered in several previous works [33,32,2]. They offer the advantage to benefit from computationally efficient solutions (e.g. Dijkstra) to optimize a sequence of ball candidates associations with respect to a desired cumulative cost over each association. They can thus compute globally optimal solutions, where the consensus-based approximations only explore a subset of the trajectory space. Since our work in [2] is the only one to directly embed the ballistic constraint in a graph-based formulation, we adopt it as a reference in our comparison with consensus-based solutions.

[6] False detections generally appear in bursts, and candidate location inaccuracies result from calibration and synchronization errors in multi-view set-ups.

The approach is applied on the serie of temporal windows $[k.T/2,(k+2).T/2]$, $k \geq 0$, and considers two graphs.

Position graph: The position graph $\mathcal{G}_\mathcal{P} = (\mathcal{N}_\mathcal{P}, \mathcal{E}_\mathcal{P})$ is a directed graph, where:

- $\mathcal{N}_\mathcal{P}$ is the set of nodes. The i-th node corresponding to the i^{th} ball candidate in the investigated time window, and being defined by a time t_i and a 3D position $\mathbf{x}_i$.
- $\mathcal{E}_\mathcal{P}$ defines the connectivity between the ball candidates. An edge is created from node i to node j when $t_i < t_j < t_i + T_{max}$ and $||\mathbf{v}_{i,j}|| \leq v_{max}$, with $\mathbf{v}_{i,j} = (\mathbf{x}_j - \mathbf{x}_i)/(t_j - t_i)$ and $||.||$ denoting the l^2-norm.

Creating a connectivity on a T_{max} time horizon increases robustness against missed detections, while upper bounding the velocity $\mathbf{v}_{i,j}$ between two nodes reflects the limited energy of a thrown ball.

Velocity graph: The velocity graph $\mathcal{G}_\mathcal{V} = (\mathcal{N}_\mathcal{V}, \mathcal{E}_\mathcal{V})$ is the line graph $L(\mathcal{G}_\mathcal{P})$ of the position graph. Hence [44]:

- $\mathcal{N}_\mathcal{V}$ is defined to be the set of edges of $\mathcal{G}_\mathcal{P}$
- $\mathcal{E}_\mathcal{V}$ is defined so that two nodes in $\mathcal{N}_\mathcal{V}$ are connected if and only if their corresponding edges in $\mathcal{G}_\mathcal{P}$ share a common node, i.e.

$$\mathcal{E}_\mathcal{V} = \{e_{i,j,k} \mid e_{i,j} \in \mathcal{E}_\mathcal{P} \ \& \ e_{j,k} \in \mathcal{E}_\mathcal{P}\}_{i,j,k \in [1,N]}. \quad (7)$$

An acceleration vector $\mathbf{a}_{i,j,k}$ is associated to each edge $e_{i,j,k}$, and is defined as $\mathbf{a}_{i,j,k} = 2.(\mathbf{v}_{j,k} - \mathbf{v}_{i,j})/(t_k - t_i)$.

Ballistic trajectory from the velocity graph: Since a ball has a constant acceleration, looking for a ballistic trajectory is equivalent to finding path in the velocity graph for which the observed acceleration is constant and close to the gravity acceleration. As a consequence, we define the cost $c_{i,j,k}$ of an edge $e_{i,j,k}$ in the velocity graph $\mathcal{G}_\mathcal{V}$ as:

$$c_{i,j,k} = (t_k - t_i).||\mathbf{a}_{i,j,k} - \mathbf{g}||, \quad (8)$$

so that the shortest path in the velocity graph corresponds to a sequence of ball candidates that follow a close-to-ballistic trajectory. In practice, the shortest paths and their associated costs $c_{i,j,k,l}$ between all vertices $e_{i,j}$ and $e_{k,l}$ (with $t_j \leq t_k$) in the velocity graph $\mathcal{G}_\mathcal{V}$ can be computed with Dijkstra's algorithm. As the time duration of those paths are different, the path costs $c_{i,j,k,l}$ are normalized and given by:

$$\widetilde{c_{i,j,k,l}} = \frac{c_{i,j,k,l}}{(t_k + t_l - t_i - t_j)^\beta}, \quad (9)$$

where $\beta > 1$ allows to favor long (in terms of time duration) shortest paths.

Paths shorter than a threshold (τ_{short}) are discarded. If more than one sufficiently long path exist, the one with the smallest normalized cost is selected if this cost is below a threshold $\tau_{cost,Graph}$. The thresholds are relatively easy to tune, according to the common duration of a ball trajectory in the sport at hand.

5 Experimental validation

In the previous sections, we have extended our work in [1], both in terms of foreground mask definition, and of masks fusion in the multi-view case. We have also observed that the previous literature considers either model fitting or graph formulation to deal with the estimation of ballistic trajectory segments. Hence, our validation primarily aims at (i) demonstrating the benefit of our contributions compared to our initial formulation in [1], and (ii) positioning our proposed consensus-based approach with respect to a state-of-the-art graph-based approach, such as the one we had introduced in [2]. Our validation also discusses the advantages of our approach in terms of real-time implementation.

To run those validations, we have augmented the APIDIS dataset with manually annotated ball positions that are now made publicly available at [24], with most of the code associated to our proposed method. The APIDIS dataset consists in 7 streams of 1600×1200 images captured by 7 calibrated IP cameras, at about 25 frames per second. During the acquisition, a timestamp has been appended to each image by a central server, allowing for a loose synchronization of the streams. To assess and compare the ball detection algorithms, a 3D ball positions ground truth has been computed based on the triangulation of the 2D ball positions manually provided in each camera view, for a 3 minutes-long period.

The comparative investigation associated to this novel and relevant[7] dataset demonstrates the superiority of the players-aware foreground mask compared to an uncleaned mask, as well as the benefit of a non-linear fusion of the correlations computed in different views. It also reveals that the RANSAC-based approach for ballistic trajectory analysis favorably compares with graph-based alternatives, while giving the opportunity to handle the cost-effective single view case. In terms

[7] The dataset is unique in terms of relevance because (i) the camera placement has been selected for production purpose rather than to detect the ball, and (ii) the lightning is far from optimal[26,40], but representative of what is commonly encountered in practice.

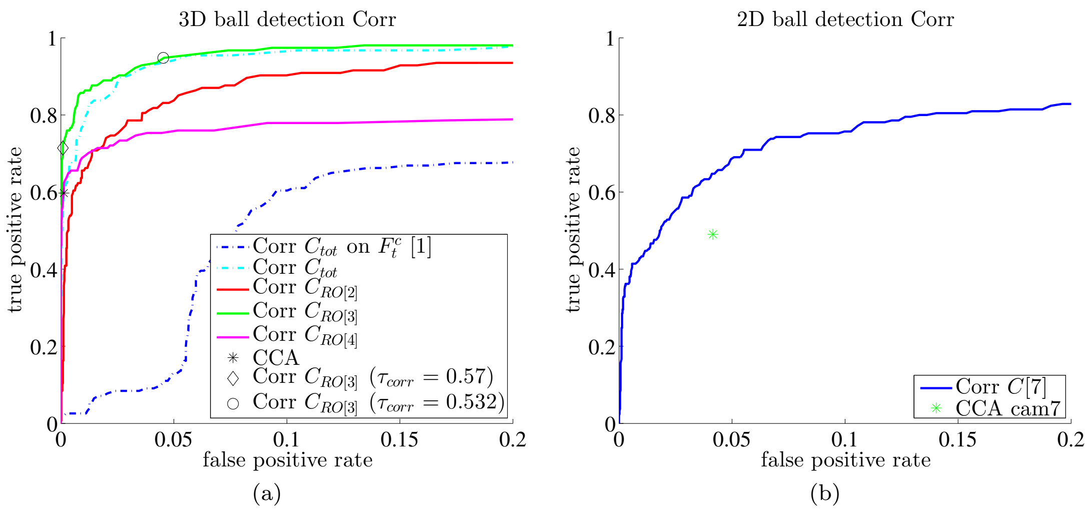


Fig. 4 (a) ROC curves for the 3D ball detection in a multi-view set-up. $N = 4501$ and $D = 154$. The dashed-dotted blue curve depicts the results obtained in [1], while other curves refer to variants of our proposed approach (see text for details), (b) ROC curve obtained when detecting the ball from a single camera with the Corr method, using the normalized correlation, and CCA operating point. In the single view case, $N = 4501$ and $D = 210$.

of computational complexity, working based on foreground mask integral images is shown to achieve real-time performance at 30 fps while using only a small fraction of a conventional CPU and associated data transfer capacities. This makes it an attractive building block for advanced team sport analysis and autonomous production. Regarding the real-time aspects, our validation also discusses the impact of the delay required to estimate the ball trajectory. It reveals that a delay of one second is largely sufficient to get the best out of the consensus-based approach, which is compatible with most real-life applicative scenarios, including automatic production [6–8], where a delay of 2 seconds is needed to smooth out the viewpoint selection decisions.

A number of video samples are provided at [45], for qualitative assessment of our method.

5.1 Detection at independent time instants

This section compares the Corr and CCA methods, presented in Figure 2 to select ball candidate positions at individual time instants. It compares the approaches in the multi- and single-view scenario. Its purpose is to quantify the benefit resulting from the contributions we have proposed to improve [1], but also to identify in which acquisition scenarios the detection on independent frames fails and needs to be augmented based on trajectory analysis. In practice, $K(= 7)$ loosely synchronized and calibrated IP cameras are considered in the multi-view scenario. For the single-view case, only camera 7 has been considered. The results have been computed over three minutes of the game starting at 18h47'00", that is for 4501 frames.

Results are presented in the form of ROC curves. Since our algorithms target the visual detection of ball trajectory segments (without complementary semantic high-level reasoning, as discussed in our 'related work' section), the detection rate corresponds to the number of true positive detections among the D time instants at which the ball (i) is not in a player's hand, and (ii) lies within the investigated field or basket area. The false positive rate corresponds to the number of false detections among the total number $N = 4501$ of investigated time instants. D is defined in the figures' legends, since it depends on the size of the investigated area.

The algorithms parameters are defined as follows. Triangulation in the CCA method considers a parameter τ_{close} equal to 30 cm, while the Corr method considers a 3D grid with a 10 cm step. Detection is considered as being valid if its distance to the ground truth is smaller than 30 cm (account for radius of the ball and error of synchronization between different cameras). τ_{big} and τ_{small} are set to 2500 and 50, respectively.

Figure 4-(a) considers a multi-view set-up, and restricts the investigated area to a rectangular box around the basket. Restricting the search to the basket neighborhood not only limits the computational complexity, but also makes sense from an applicative perspective. Indeed, as pointed out by the work in [46], knowing when the ball is close to the basket is of primary importance to disambiguate the game actions. In our dataset, the basket area is covered by 4 cameras, and Figure 4-(a) compares the ROC curves resulting from [[1], p. 1,

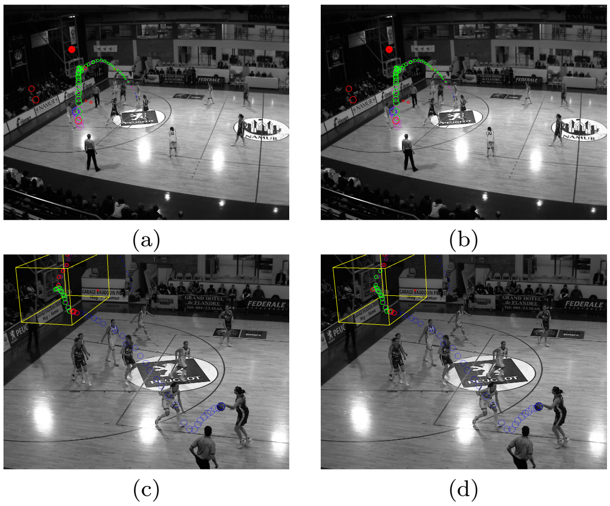


Fig. 5 Ball detection process illustration. Video available at [45]. Blue and purple circles depict the ground truth of the ball. Blue circles correspond to the instants at which the ball cannot be detected by our methods, either because it is in the hand of a player, or it lies outside the investigated area. Green and red circles depict the 3D ball candidates that are detected at individual time instants. Green circles identify the positions that are found to be consistent with a ballistic model. (a) and (b) are obtained with the CCA method, filtered based on a graph or RANSAC, respectively. (c) and (d) are obtained with the Corr method, with graph-based and RANSAC filtering, respectively.

eq. (3)], with the curves resulting from the methods described in Section 3.2. Those methods compute the correlations with respect to the players-aware cleaned-up difference of consecutive foreground masks and not on the foreground mask itself as in [1]. They then merge those correlations based on a summation (C_{tot}) or a r^{th} order rank filtering ($C_{RO[r]}$), $2 \leq r \leq 4$[8]. We observe that, by working on the difference of two consecutive foreground masks, we are less sensitive to noise, and we achieve better results than working on the foreground mask F_t^c (see blue and cyan dash-dotted curves). We also observe that the best results are obtained with the third order rank ordering filter. The second order filter is too sensitive to noise, which induces many false positives, while the fourth order (=maximum, since only 4 views are available) filter is too conservative and fails in detecting many candidates[9]. Interestingly the CCA operating point lies close to the best Corr ROC curves, which means that CCA provides a reasonable alternative when the large size of the investigated area makes Corr impractical in a real-time context. We however observe that the CCA ends up in a relatively small detection rate. This is a potential drawback since we show in the next section that the trajectory analysis helps in preserving the correct detections while sharply reducing the false positive rate. Hence, it might be judicious to run the ball candidates detector at an operating point that results in a high detection rate, even if the amount of false positives increases (see dashed lines in Figure 6(b)).

Figure 4-(b) considers ball detection in a single-view set-up, also around the basket area. Specifically, it shows the CCA operating point and the ROC curve obtained with the Corr approach, using the normalized correlation in camera 7. We observe that the false detection rate becomes significant as long as the detection rate increases beyond 50%. Hence, ballistic trajectory analysis becomes relevant to discriminate between false and true positives in the single-view case.

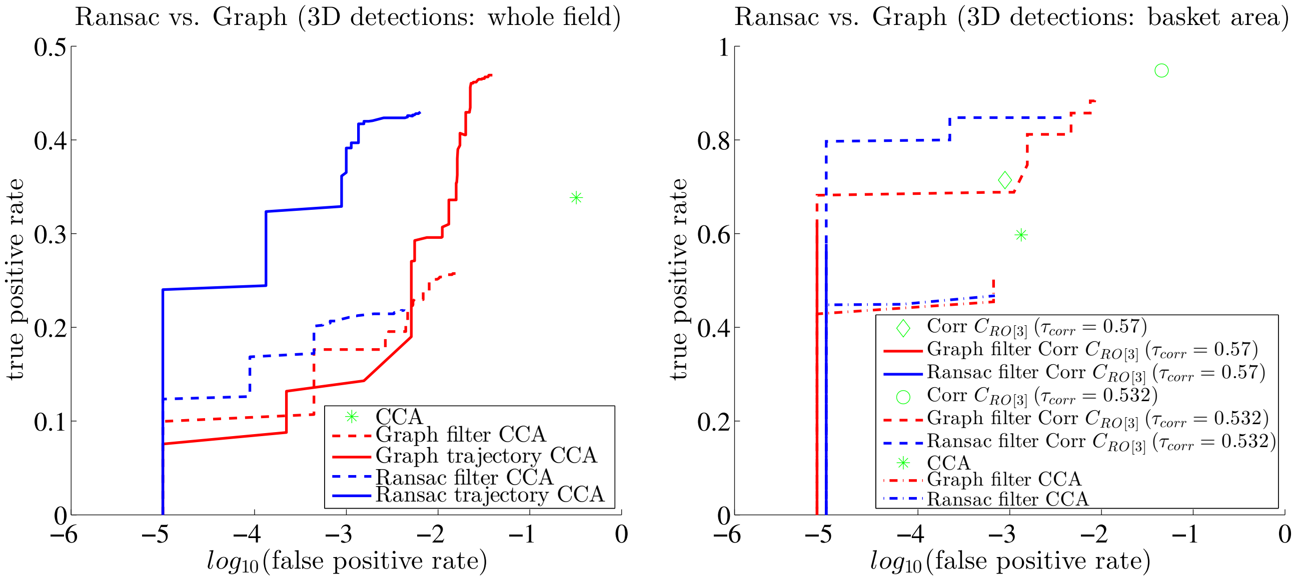
In a multi-view set-up, the detection accuracy obtained in the basket area and presented in Figure 4(a) is sufficient, and does not justify a trajectory analysis. However, this is not the case anymore when the whole field is considered, as illustrated in Figure 6-(a). In this figure, we observe that the CCA operating point drops from 70-60% when considering the area around the basket, to 33.7% when considering the whole field. This might sound like a poor detection rate, but it can be explained since, even if the ball is not in hands, it can be impossible to detect it using our approach because its silhouette is lost in players silhouettes (during a dribble for example). The relevance of our detection rate is confirmed by the videos available at [45], where we observe that the ball is most often at least sporadically detected when lying between the players (during a pass) or moving towards or from the basket (during a shot)[10]. The presence of false positives, which could potentially lead to erroneous game interpretation [44], is not dramatic at this stage since the next section reveals that ballistic trajectory analysis both erase them, and bridge the gap between sporadic true positives.

Before moving to the ballistic trajectory analysis, we conclude from this section that, when it is computationally affordable, template-based correlation should be preferred to heuristic connected region analysis such as the one envisioned in [2]. We also observe that our proposed correlation-based approach significantly improves the one initially presented in [1]. Eventually, we are interested in ballistic trajectory analysis when the

[8] For some M, the order r might be larger than the number of views in which M is visible. In that case, the order of the filter is set to the number of available views.

[9] Note that the first rank order filter has not been depicted because it leads to poor performance, due to the noisy nature of individual foreground masks, but also due to the fact that a single view 2D location is associated to many 3D positions.

[10] Detecting the ball in those situations is important regarding sport analysis because it complements the cues derived from players displacements (see [26] and [44]).



(a) (b)

Fig. 6 (a) ROC curves obtained after graph-based and RANSAC ballistic analysis of the ball position candidates detected by the CCA method on the whole basket-ball field, for which $N = 4501$ and $D = 1616$. The term 'filter' refers to the rejection of candidates that do not follow a ballistic trajectory, while the term 'trajectory' refers to the interpolation of missed detection in addition to non-ballistic samples rejection. (b) ROC curves obtained after ballistic analysis of the ball position candidates detected by the Corr and CCA methods around the basket. Graph-based [2] and RANSAC methods are compared. Note: false positive rate is artificially bounded to -5 in logarithmic scale.

detection at independent time instants implies significant false positive rates at satisfying detection rates. This case is the most common in practice.

5.2 Ballistic trajectory analysis

Ballistic trajectory analysis is illustrated in Figure 5. It essentially helps in case of significant false positive rates, as primarily encountered in the single view case (see Fig. 4-(b)), or when dealing with the whole basket-ball field in the multi-view set-up (see Fig. 6-(a)). In the multi-view case, our validation shows that our proposed RANSAC approach compares favorably to the graph-based approach introduced in [2]. It then demonstrates that the extension of this RANSAC method to a projected ballistic model makes it possible to reliably detect the ball with a single camera.

The parameters of the graph-based algorithms have been set as follows (see [2]): $T_{max} = \tau_{short} = 0.2$ sec, $v_{max} = 25$ m/sec, $\beta = 3$, and $\tau_{cost,Graph} = 0.5$.

Regarding the RANSAC approach, the model with minimal mean squared error is only kept if the error is smaller than a threshold $\tau_{cost,RANSAC}$, set to 6 centimeters.

In the multi-view scenario, Figures 6-(a) and 6-(b) present the ROC curves obtained after ballistic analysis of CCA and Corr results, respectively when considering the whole field and the basket area. Those figures compare the results obtained either with the graph-based or the RANSAC approach (mean of ten trials). The 'filter' curves refer to the performance obtained by filtering out the candidates that do not fit a ballistic model, while the 'trajectory' curves compute the performances by interpolating missed detections between the inliers, i.e. the candidate positions that follow a ballistic model, in addition to rejecting outliers. Figures 6-(a) and 6-(b) show that (i) ballistic filtering is quite effective in removing false positives, (ii) the graph-based and the RANSAC approaches achieve comparable results, and (iii) ballistic interpolation of missed detection substantially increases the detection rate (by 10-15%). The RANSAC approach has the advantage to naturally extends to single-view, as illustrated in Figure 7, which presents the ROC curves obtained when detecting ball position candidates from a single camera view, using the CCA method. It shows that a RANSAC filtering removes most of the false positive ball detections, while preserving a large majority of true positive detections. Hence, it becomes possible to detect the ball when it is close to the basket, even with a single camera. This is of primary importance regarding game interpretation, in the context of autonomous production [6,46] or team activity recognition [47].

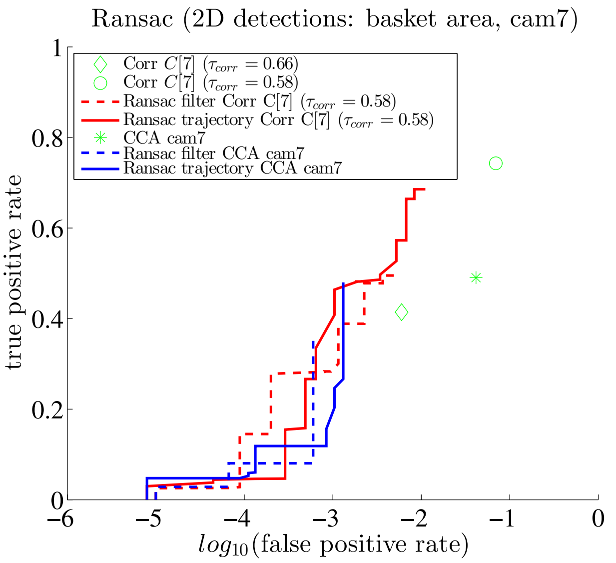


Fig. 7 ROC curves obtained in the single view case (Camera 7 in APIDIS dataset). The CCA and Corr methods are considered to compute the ball candidates. The RANSAC filtering is based on a projected ballistic model (see Section 4.1.2).

5.3 Real-time implementation aspects

Our algorithm has been designed to minimize the computational complexity. In a real-life system, ball detection is supposed to be combined with a player detection algorithm that also relies on a foregound mask [4, 3]. Hence, the marginal complexity associated to the ball detection is mainly due to the computation of the integral image of the cleaned-up mask in spherical coordinates, and to the computation of the correlations with the projected silhouette templates associated to the whole set of ball position candidates[11]. Only 7 additions/subtractions (=binary template matching) and one multiplication (=normalisation) have to be computed to estimate the correlation in one ball position candidate. This is at least 2 orders of magnitude smaller than for color histograms inner products (e.g. as used in [26]), including when those ones are implemented using integral histograms [48]. As a consequence, our system is expected to be data-dominated, meaning that its throughput is determined by the transfer of data rather than by the computational load.

To quantify the computational requirements of our method without being limited by the speed at which the input frames are fowarded by the camera or read from the hard drive, we have run our system while continuously and iteratively accessing the same input image (from the cache). In this experiment, each processing iteration computes a rectified foreground mask for player detection (see [4]), and maps it to spherical coordinates for ball detection (see Section 3.2.1). In our case, the 1624×1234 input image captures half of the sport field (as in the videos provided for qualitative assessment [45]). It is sub-sampled by two in both directions, before rectification (using a look-up table) and foreground mask computation. Player template integral computations are then performed on a 640×480 image (=part of the rectified image that covers the sportfield) to feed the ground occupancy map, defined by sampling the sportfield with a 4 cm squared grid. The ball detection is run around the basket (within a 4m × 3m × 3m box, over a regular grid sampling candidates every 10 cm in the 3 directions), which means that 36000 ball template correlations are computed at each time instant. The spherical coordinate system considered for ball detection is centered in the camera location, and its X-axis points towards the basket. The image used for ball template integrals computation is then obtained by sampling (using a look-up table) the foreground mask symetrically with respect to the X-axis, both in azimuthal (-8 to +8 degrees) and polar (+84 to +96 degrees) angles. The sampling rate is set to 10 degree per samples so as to preserve the resolution of the input foreground mask. The processor used integral image computations was a hyper-threaded (meaning two logical processor per core) quad-core i7-4790 CPU @ 3.60GHz. Our implementation builds on a linux operating system (Ubuntu 14.04 LTS, 64 bits), and uses Intel® Integrated Performance Primitives (IPP), and C/C++ programming language. The observed throughput over 10 trials was 12053 iterations/sec, with a standard deviation of 122 iterations/sec. Table 1 presents the percentage of resources used by each component of the system. This percentage is expressed in terms of logical processor usage. Hence, when fully activated, the 8 threads of the processor correspond to 800% of resources. We observe that none of the processes saturates one of the logical processor, which confirms that computations do not constraint the throughput.

System component	Logical processor usage (%)
Foreground mask	37 %
Look-up and integrals	41 %
Player detection	60 %
Ball detection	56 %

Table 1 Percentage of logical processor (hardware thread) utilization associated to a players and ball detection system. Total resources for the tested quad-core processor correspond to 800% of virtual/logical processor. Input image is accessed from fast cache memory. System runs at 12000 frames/sec.

[11] We neglect the complexity associated to the RANSAC-based ball trajectory estimation, since it runs over a relatively small number of pre-selected ball position candidates.

Predicting the capacity in other usage scenarios (with more camera viewpoints, or whole field ball detection) is not trivial since, in multi-thread/multi-core CPU architectures, capacity extrapolation requires to know both the detailed utilization information of all shared core processing units (arithmetic and logic unit, floating point unit, cache, memory bandwidth, etc.) in the reference usage scenario, as well as the characteristics of the workload to be added in the targeted scenario. However, since the processor usages provided in Table 1 are constrained by the data transfers and not by the computational resources, at least as many operations will be available when working at smaller frame rates. Hence, at the conventional rate of 30 frames/sec, we can reasonably consider that at least 14×10^6 ball template correlations can be investigated per frame[12], whatever the acquisition set-up.

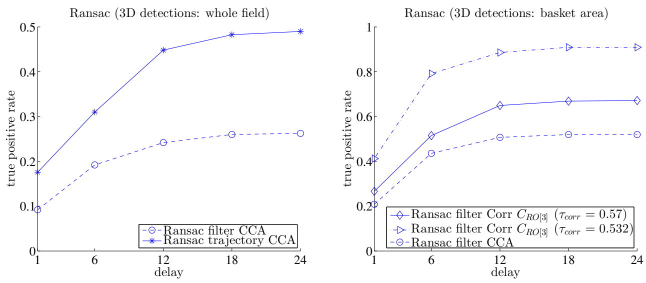
Table 2 considers a variety of acquisition scenarios, and presents the number of template correlations and the associated percentage of logical processor resources, as estimated for the different scenarios running on a quad-core i7-4790 at 30 frames/sec. All scenarios search the ball over a regular lattice of vertices, the distance between two adjacent vertices being equal to 10 cm. The first two rows consider a multiview acquisition set-up, in which 4 cameras are considered to cover each side of the field from 2 distinct viewpoints. The first row only searches the ball around the basket area, in a 4m × 3m × 3m box, while the second row considers a search over the whole 15m × 30m field, over a 4m height. The last two rows consider a single viewpoint acquisition set-up, with whole field search. The third row considers the conventional template correlation approach described in Section 3.2.1, while the last row considers the approximation presented in Section 3.2.3, where the panoramic image is scanned with a constant-size sliding template. When covering the whole field, the spherical system used for ball template integrals computation points towards the center of the field and samples the foreground mask derived from two cameras (one for each side of the field) at 10 samples per degree symetrically with respect to the X-axis, both in azimuthal (-60 to +60 degrees) and polar (+60 to +120 degrees) angles. The size of the spherical image covering the field from a side view is thus 1200 × 600. The correlation has to be computed every two samples to maintain the distance between adjacent templates below 10cm, whetever the depth of the ball in the scene. Hence, 600 × 300 correlations have to be computed in every fame. We observe that, at 30 fps, the detection of the ball can be implemented in real-time with very small computational overhead, especially when only the basket area is considered, or when the constant-size template approximation is acceptable (as it is typically the case when dealing with panoramic side views) and supports a sliding window implementation, as explained in Section 3.2.3 .

[12] 14×10^6 = 12000 fps × (40 × 30 × 30) / 30 fps.

Scenario	**# corr/frame**	**Logic. proc. usage**
2 VPs, basket	0.14 M	1 %
2 VPs, field	3.6 M	26 %
1 VP, field	1.8 M	13 %
1 VP, field, SW	0.18 M	1.3 %

Table 2 Number of template correlations per frame, and estimation of the associated percentage of logical processor utilization on a quad-core i7-4790 running at 30 frames/sec, in a variety of applicative scenarios. Each scenario involves either 1 or 2 viewpoints (VPs), and investigates either the whole basket-ball field or just the basket area, as detailed in the text. SW refers to the sliding window approximation introduced in Section 3.2.3.

In Figure 8, we go one step further and investigate the impact of the delay parameter d, and associated buffering requirements, on the ballistic trajectory analysis. We observe that, at constant false positive rate, the detection rate does not much improve anymore beyond a delay of about 12 frames, but degrades significantly when it becomes smaller than 12, which corresponds to about half a second. This reveals that ballistic trajectory analysis can only support delay-tolerant applications, and would for example not be suited to drive motorized cameras in an instantaneous and real-time autonomous production scenario.



(a) (b)

Fig. 8 Impact of the delay parameter d in both cases (a) when the ball position candidates is detected by the CCA method on the whole basket-ball field and (b) when the ball position candidates is detected by the Corr and CCA methods around the basket. The detections are filtered with the RANSAC method, the size of the window is fixed to $T = 24$, the shift to $s = 1$ and the false positive rate to 0.001.

6 Conclusion

Our investigations have demonstrated that reliable and cost-effective detection of a weakly contrasted ball can be supported in cluttered sport scenes by single-view acquisition systems, as long as advanced image and signal processing tools are implemented to cope with the limitations of conventional and error-prone foreground silhouettes detectors. Non-linear rank-order filters have been proposed to effectively merge the foreground cues extracted from multiple views based on the correlation with a ball silhouette template, and a RANSAC-based solution has been investigated to extract ballistic trajectory samples out of the error-prone sequences of ball position candidates detected by a foreground detector. Our work has shown that the RANSAC-based solution is competitive compared to alternative graph-based solutions, while offering the additional advantage to naturally extend to the cost-effective single-view scenario. Our work also helps in making important design choices. Multiple views significantly increase the detection accuracy, and panoramic views are recommended since they observe the whole scene from a far viewpoint, which makes the size of the ball template independent of the actual position of the ball on the field. In the single view case or when detection is desired over the whole cluttered scene, ballistic trajectory analysis is recommended, but requires a minimal delay (of about half a second) to be effective.

Acknowledgements

Part of this work has been funded by the Belgian NSF, and the Walloon region projects SPORTIC and PTZ-PILOT.

References

1. A. K. K.C., P. Parisot, C. De Vleeschouwer, Demo: Spatio-temporal template matching for ball detection, in: ICDSC, 2011.
2. P. Parisot, C. De Vleeschouwer, Graph-based filtering of ballistic trajectory, in: ICME, 2011.
3. P. Parisot, B. Sevilmiş, C. De Vleeschouwer, Training with corrupted labels to reinforce a probably correct teamsport player detector, in: ACIVS, 2013.
4. D. Delannay, N. Danhier, C. De Vleeschouwer, Detection and recognition of sports (wo)men from multiple views, in: ACM/IEEE ICDSC, 2009, pp. 1–7.
5. I. Fernandez, F. Chen, F. Lavigne, X. Desurmont, C. De Vleeschouwer, Browsing Sport Content through an Interactive H.264 Streaming Session, in: MMEDIA, 2010, pp. 155–161.
6. F. Chen, D. Delannay, C. De Vleeschouwer, An autonomous framework to produce and distribute personalized team-sport video summaries: a basket-ball case study, IEEE Trans. on Multimedia 13 (6) (2011) 1381–1394.
7. F. Chen, C. De Vleeschouwer, Personalized production of basketball videos from multi-sensored data under limited display resolution, CVIU 114 (6) (2010) 667–680.
8. Keemotion, Keemotion production technology, http://www.keemotion.com.
9. L. Sun, Q. De Neyer, C. De Vleeschouwer, Multimode Spatiotemporal Background Modeling for Complex Scenes, in: EUSIPCO, 2012, pp. 165–169.
10. C. Stauffer, W. Grimson, Adaptive background mixture models for real-time tracking, in: CVPR, 1999, pp. 246–252.
11. T. Horprasert, D. Harwood, L. Davis, A statistical approach for real-time robust background subtraction and shadow detection, in: ICCV, 1999, pp. 1–19.
12. V. Reddy, C. Sanderson, B. Lovell, Improved foreground detection via block-based classifier cascade with probabilistic decision integration, IEEE TCSVT 23 (1) (2013) 83–93.
13. J. Zhou, J. Hoang, Real time robust human detection and tracking system, in: CVPR, 2005.
14. A. Cavallaro, O. Steiger, T. Ebrahimi, Semantic video analysis for adaptive content delivery and automatic description, IEEE TCSVT 15 (10) (2005) 1200–1209.
15. S. Khan, M. Shah, A multiview approach to tracking people in crowded scenes using a planar homography constraint, in: ECCV, Vol. 4, 2006, pp. 133–146.
16. F. Fleuret, J. Berclaz, R. Lengagne, P. Fua, Multi-camera people tracking with a probabilistic occupancy map, IEEE Trans. on PAMI 30 (2) (2008) 267–282.
17. A. Alahi, L. Jacques, Y. Boursier, P. Vandergheynst, Sparsity driven people localization with a heterogeneous network of cameras, Jour. of MIV 41 (1-2) (2011) 39–58.
18. P. Dollar, C. Wojek, B. Schiele, P. Perona, Pedestrian detection: a benchmark, in: CVPR, 2009.
19. P. Felzenszwalb, R. B. Girshick, D. McAllester, D. Ramanan, Object detection with discriminatively trained part-based models, IEEE Trans. on PAMI 32 (9) (2010) 1627–1645.
20. P. Viola, M. Jones, Robust real-time object detection, in: Int. workshop on SCTV, 2001.
21. Hawk-Eye, Hawk-Eye ball tracking system, http://www.hawkeyeinnovations.co.uk/page/sports-officiating.
22. F. Yan, W. Christmas, J. Kittler, A tennis ball tracking algorithm for automatic annotation of tennis match, in: BMVC, 2005, pp. 619–628.
23. K. Choi, Y. Seo, Probabilistic tracking of the soccer ball, in: ECCV, 2004, pp. 50–60.
24. Public, Apidis dataset, ball detection ground-truth, and code samples, http://sites.uclouvain.be/ispgroup/index.php/Softwares/APIDIS.
25. Y. Zhang, H. Lu, C. Xu, Collaborate ball and player trajectory extraction in broadcast soccer video, in: International Conference on Pattern Recognition, 2008.
26. X. Wang, V. Ablavsky, H. Ben Shitrit, P. Fua, Take your eyes off the ball: Improving ball-tracking by focusing on team play, CVIU 119 (2014) 102–115.
27. X. Wang, E. Tretken, F. Fleuret, P. Fua, Tracking interacting objects optimally using integer programming, in: ECCV, 2014.
28. C. Lampert, J. Peters, Real-time detection of colored objects in multiple camera streams with off-the-shelf hardware components, Journal of Real-Time Image Processing 7 (1) (2012) 31–41.
29. J. Ren, J. Orwell, G. Jones, M. Xu, Real-Time modeling of 3D soccer ball trajectories from multiple fixed cameras, IEEE Transactions on Circuits and Systems for VideoTechnology 18 (2008) 350–362.

30. M. Labayen, I. Olaizola, N. Aginako, J. Florez, Accurate ball trajectory tracking and 3D visualization for computer-assisted sports broadcast, Multimedia Tools and Applications 73 (3) (2014) 1819–1842.
31. H.-T. Chen, M.-C. Tien, Y.-W. Chen, W.-J. Tsai, S.-Y. Lee, Physics-based ball tracking and 3d trajectory reconstruction with applications to shooting location estimation in basketball video, Journal VCIR 20 (3) (2009) 204–216.
32. F. Yan, A. Kostin, W. Christmas, J. Kittler, A novel data association algorithm for object tracking in clutter with application to tennis video analysis, in: CVPR, 2006, pp. 634–641.
33. X. Zhou, Q. Huang, L. Xie, S. J. Cox, A two layered data association approach for ball tracking, in: ICAPSS, 2013, p. 5.
34. H.-T. Chen, W.-J. Tsai, S.-Y. Lee, J.-Y. Yu, Ball tracking and 3D trajectory approximation with applications to tactics analysis from single-camera volleyball sequences, Multimedia Tools and Applications 60 (3) (2012) 641–667.
35. B. Chakraborty, S. Meher, A real-time trajectory-based ball detection-and-tracking framework for basketball video, Journal of Optics 42 (2) (2013) 156–170.
36. F. Poiesi, F. Daniyal, A. Cavallaro, Detector-less ball localization using context and motion flow analysis, in: ICIP, 2010.
37. K. Kim, M. Grundmann, A. Shamir, I. Matthews, J. Hodgins, I. Essa, Motion fields to predict play evolution in dynamic sport scenes, in: IEEE Conference on Computer Vision Pattern Recognition , 2010.
38. Y. Ohno, J. Miura, Y. Shirai, Tracking players and estimation of the 3D position of a ball in soccer game, in: International Conference on Pattern Recognition, 2000.
39. M. Leo, N. Mosca, P. Spagnolo, P. Mazzeo, T. DOrazio, A. Distant, Real-time multi-view analysis of soccer matches for understanding interactions between ball and players, in: ACM International Conference on Image and Video Retrieval, 2008.
40. A. Maksai, X. Wang, P. Fua, What Players do with the Ball: A Physically Constrained Interaction Modeling, in: CVPR, 2016.
41. P. Viola, M. Jones, Robust real-time object detection, IJCV 57 (2) (2004) 137–154.
42. SportVU, STATS SportVU technology, http://www.stats.com/sportvu/sportvu.asp.
43. M. A. Fischler, R. C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Commun. ACM 24 (6) (1981) 381–395.
44. J. T. Gross, J. Yellen, Graph Theory and Its Applications, Boca Raton, FL: CRC Press, 1999.
45. Videos, Ball tracks.Http://sites.uclouvain.be/ispgroup/index.php/Research/BallTrac.
46. F. Chen, D. Delannay, C. De Vleeschouwer, Multi-sensored vision for autonomous production of personalized video summaries, in: Int. ICST Conference on User Centric Media, 2010.
47. M. Perše, M. Kristan, S. Kovačič, G. Vučkovič, J. Perš, A trajectory-based analysis of coordinated team activity in a basketball game, CVIU 113 (2009) 612–621.
48. F. Porikli, Integral histogram: a fast way to extract histograms in Cartesian spaces, in: IEEE Conference on Computer Vision Pattern Recognition , 2005.