Scene text recognition and tracking to identify athletes in sport videos

Stefano Messelodi, Carla Maria Modena
FBK-irst, Via Sommarive 18, I-38123 Povo, Trento, Italy

messelod@fbk.eu, modena@fbk.eu

Abstract

We present an athlete identification module forming part of a system for the personalization of sport video broadcasts. The aim of this module is the localization of athletes in the scene, their identification through the reading of names or numbers printed on their uniforms, and the labelling of frames where athletes are visible. Building upon a previously published algorithm we extract text from individual frames and read these candidates by means of an optical character recognizer (OCR). The OCR-ed text is then compared to a known list of athletes' names (or numbers), to provide a presence score for each athlete. Text regions are tracked in subsequent frames using a template matching technique. In this way blurred or distorted text, normally unreadable by the OCR, is exploited to provide a denser labelling of the video sequences.

Extensive experiments show that the method proposed is fast, robust and reliable, out-performing results of other systems in the literature.

Keywords embedded text detection; text tracking; sport video analysis; athlete identification; text reading; information extraction

1 Introduction

Since text conveys semantic information, the reading of text in images and videos plays an important role in the video content understanding process.

According to Lienhart [13], we can divide text's appearance in videos into two macro categories: overlaid text and scene text. The former group includes text that is superimposed over the images, like timestamps, captions in video news, titles of movies. As the text is deliberately superimposed, it is expected to respect certain 'visibility' rules regarding size, position, appearance, motion, in order to be readable by the viewer. On the contrary, scene text is inherently embedded within the scene, for example hotel or shop placards, road signs, street names, posters. Due to this natural presence, such text can manifest itself in a wide range of conditions, depending upon several factors related to the scene and the acquisition process. This fact, in general, makes its detection and reading a very challenging task. Reading text embedded in natural scenes plays an important role in several applications, such as the indexing of multimedia archives [25], recognizing signs in driver assisted systems [31], providing scene information to visually impaired people [7], identifying vehicles by reading their license plates [8]. Moreover, the explosion and widespread diffusion of low-priced digital cameras and mobile phones endowed with good quality cameras [6], has meant that text extraction from camera-captured scenes has gained a renewed attention in computer vision research [12, 34].

The extraction of textual information from scenes (still images or videos) can be divided into sub-stages, not all necessarily present nor quite distinct: (1) existence or absence of text in the image, (2) its localization, (3) tracking, (4) enhancement, (5) extraction, and (6) interpretation. An interesting and detailed survey of these sub-stages (excluding character recognition) is provided by Jung et al. [9]. In their paper, they categorize text localization methods into two main types: region-based and texture-based, according to the features utilized. The first group is further divided into connected component and edge-based approaches. The method [16] to extract text from images, by which the present work is inspired, is also reviewed in their paper in the connected component-based section. In some recent approaches, scene text segmentation relies upon graphical models and belief propagation [27], methods that are also interestingly applied to text recognition [30].

The module described in this paper is part of a platform developed inside the My eDirector 2012 project [22], whose main goal is to enable end-users to set-up their own coverage of large athletic events. The user, following her personal preferences, can interactively select actors/events within real-time broadcasted scenes coming from a multi-camera environment. A simplified overview of the architecture is provided in [20], while the integration of ath-

letes identification by text reading with modules devoted to detection and tracking using different cues, e.g. face and uniform appearance, is provided in [21]. Our module not only permits to identify framed athletes, but it is also used as a trigger to initiate new tracks in a more general appearance tracking system. Furthermore, the parallel comparison of results provided by the text reader on simultaneous sequences of the same event permits to infer some valuable information on the cameras' framing.

The proposed method is, as far as we know, the first one that flexibly identify athletes by their name or number on the bib within real-time broadcasted videos. In fact, it is highly flexible by exploiting the available external knowledge about the text to be searched for (the dictionary and roughly the text appearance), which allows the system to successfully process sequences in which athlete identifiers present different visual aspects. With respect to published works, we have tested the method on a very large data-set (about 250 000 frames), including shots where multiple athletes are simultaneously framed. In spite of this additional difficulty, our method out-performs previous published methods. Furthermore, the proposed implementation is suitable for real-time elaboration of high resolution (1024×576) videos at 25 fps.

This paper is organized as follows: Section 2 focuses on the analysis of text in sport images, describing the problems, pointing out the major difficulties, discussing the exploitation of prior knowledge and presenting some related works. Section 3 presents an outline of our method, whilst each of the three subsequent sections devote themselves to the description of three main modules: text extraction, athlete identification and text/athlete tracking. The results of the experiments conducted on eight undirected athletics videos and several videoclips (for a total of 249 171 frames) are reported in Section 7. Conclusions and problems to be faced in future work are reported in Section 8.

2 Text in sport images

The problem Since the late '90s the problem of associating names to people has attracted many researchers, in particular in the video indexing field. The textual information provided by video captions, like those found in TV news, is one of the most important sources of media to annotate, and the challenge of solving the well-known face-name association task is particularly

attractive. In our sports application, textual information is not superimposed into the scene (as we are dealing with un-edited, or raw, footage), instead it is embedded inside the image, i.e. the name or identification number of athletes are attached on their uniforms.

We can clearly define two detection scenarios associated to the problem, namely: Who is the person in the scene? (having never been seen before) [26], and the so called "person X finding" problem [32], which requires us to detect which video frames contain the named person.

The task addressed in this paper belongs to the latter class, as knowledge concerning athletes' names list in a particular sport event is known a priori (domain knowledge). However, text localization and reading can also be exploited to solve the first problem (person naming). In fact, once text is located in an image, a face detector can be guided towards a region specified above the text, and if present, it can provide a text-face association. In cases where the module provides high identification confidence, it is possible to capture one (or more) instances of the athlete's face to create/update a face-appearance almanac for face recognitions in the same video. This is particularly useful in future frames where extreme close-ups push text off the bottom of the screen. Moreover, locating and identifying athletes in frames provides a useful trigger to initialize an appearance-based person tracking module. In all of these cases the core problem is the realization of a module for text localization, extraction and recognition which provides a reliable output in real-time for a personalized media streaming. Actually, the goal of the My eDirector 2012 [22] project is to provide an interactive broadcasting service enabling end-users to select focal actors within real time broadcasted scenes.

The problem can therefore be formulated in the following way: given a name, find the coordinates of video frames where this person is present.

Difficulties Mayers et al. [18] provide a classification of the difficulties which is related to the number of degrees of freedom in which text can appear in 3D space, by modelling the orientation of text relative to the camera in three angles. However, their classification only encompasses text that falls on a planar surface, whereas text often appears on a non-planar surface, e.g. on a cylinder or on a deformable surface like an athlete's jersey. This problem therefore exceeds the maximum difficulties considered in [18]. A list of the difficulties of text extraction from natural scenes is also reported in [14].

Most of these factors are present in the task at hand, such as blurring, due to both the athlete's and camera's motion, incorrect lens focus, variations in illumination with shadows and glares, non-constant resolutions of the text either far or close to the camera and complex backgrounds. Moreover, it is important to note that text present in our particular scenario often has few characters, like numbers on jerseys or short names, which presents a further level of difficulty because a common hypothesis made by text location algorithms is that text strings must have at least three characters. Text on an athlete's uniform may identify not only their name or number, but also their nationality, club membership, sponsorship, or the make of the shirt itself. The camera also captures text in the background scene, containing words like the name and location of the meeting, corporate sponsorships, advertisements, text from dynamic billboards, lane numbers, and so on. Overlaid text can also be present in edited video material (with athletes' names, record times, laps to go, etc.). Naturally, in some frames, text is not present at all.

Once text is detected and recognized, other difficulties arise related to tracking in subsequent frames. In fact, text motion can be a very complex activity to predict due to dynamic camera operations (like panning, zooming, shot framing, etc.) as well as the inherent dynamic nature of athletes (rotating, jumping, etc.).

Prior knowledge By exploiting external prior knowledge (when available), the complexity of the problem can be considerably reduced. A knowledge of competition-type helps to customize the system to improve the text localization, while a list of competing athletes can be exploited in the recognition step, by adding the strings to the OCR vocabulary and checking the results against a list of expected candidate names.

In our scenario the athletes' names (or numbers) are black on a white background. Furthermore, text alignment is near-horizontal, with a skew due to body pose. Only in sporadic cases does text appear up-side-down (Figure 1, right) or greatly sloped. Furthermore, the names on the bibs are known a priori. This knowledge is coded in external parameter files fed as input to the system.

Text on athletes: related works Many attempts to automatically annotate sport videos have been made. A survey of approaches in sports-related indexing and retrieval can be found in [10], however, relatively few works

Figure 1: Example of text on jerseys with different appearance: dark on yellow, black on white, skewed, name, number, and upside down.

address the extraction of text from athlete's jerseys for identification.

In [33], images are segmented using a generalized learning vector quantization algorithm that reduces the number of colours and assigns pixels to homogeneous regions in order to separate jersey numbers from their background. To discard non-jersey number regions, dimension, area and thickness attributes are taken into consideration. Surviving regions are represented by means of features invariant to scale and rotation (Zernike moments). A k-nearest neighbour classifier is employed to classify or discard a candidate digit (after a training phase using samples of synthetic, non-rigid, digit deformations). Candidate regions are tracked using the Sum of Squared Difference image matching algorithm described in [11] and assessed through subsequent frames. The final classification is obtained by a voting procedure. They report, for detection only, a recall of 62% and a precision of 84% computed on 200 frames with jersey numbers, while for detection and tracking together, a recall of 77% and a precision of 87% computed at frame level on 30 short videoclips.

The goal of the research in [2] is to automatically annotate soccer videos with player identities. First, faces are detected and tracked in close-up shots. Then the frames in which faces are detected are also probed to find the player's number depicted on the jersey, or for superimposed text captions. Jersey number detection is achieved through an implementation of the algorithm reported in [29], using multiple detectors, each of them trained with positive and negative examples for each specific jersey number, ranging from 1 to 22. This algorithm relies upon a number of simple classifiers, that detect the presence of a particular feature of an object to be detected. Each detector acts as a dichotomizer, additionally enabling the system to directly recognize a number. Initially, a large number of simple features are considered; a high performance classifier is then constructed by selecting a small number of im-

portant features using AdaBoost. They report a recall of 56% and a precision of 83% computed on 36 shots, with resolution 360×288, where the number is present. No performance at frame level is provided. The same authors also propose a different method in [3]. They focus on a zone of interest by firstly detecting and clustering Harris corners, and then on maximally stable extremal regions. In this way, they extract binary candidate text regions which are subsequently fed into a standard OCR. With this method, recall is 68% and precision is 84% on 40 shots (6 000 frames) where the number is present, being these figures computed using the shot as basic unit.

The approach presented in [24] relies upon an image segmentation method based on the colour contrast between number and jersey, represented by average colour vectors in the HSV colour space; these are pre-computed from examples. This results in the creation of a bitmap in which the numbers are represented as 'holes' in the jersey regions. In this way, candidates are extracted as internal contours of objects and then filtered according to area and bounding-rectangle aspect ratio. The located regions are then rotated according to the contour central moments and afterwards smoothed using median filtering. Candidate regions are subsequently grouped by distance to accommodate double digit numbers, these are then fed into an OCR. Temporal redundancy of OCR result is checked. On a set of 1116 selected frames with dimensions 640×480 or 640×352, where the player number is present, the correct localizations and identifications are 328, false alarms 51, and miss detections 788, leading to a precision of 87% and a recall of 29%. At shot level, the system outputs 28 identifications, of which 20 are correct, on a set of shots taken from a data-set with 58 430 frames, 31 of which showing player number. At shot level the precision is 71% and the recall is 65%.

Tracking techniques specific for scene text are limited in literature. In Andrade et al. [1] the sport image is firstly segmented by colour; a region adjacency graph and picture trees are constructed to isolate and track players by exploiting prior knowledge, such as the players' shirt colours. Region analysis is then applied to the focused zones to isolate players' numbers, using the knowledge of number colour and its surrounding background (the shirt). Candidate numbers are then normalized according to size and then classified into one of the digits, or else rejected. The tracking is performed in the region-space domain, by comparing the list of detected objects having similar description in the current and previous frames, where the description relies upon statistics of the region and the neighborhood. This kind of descriptor does not affect tracking in case of partial occlusions. Results of

players detection are not provided, while tracking results are given for a sport sequence of 55 frames. Performance of the digit classifier are also provided with a rejection versus error plot, where, for 0 rejections, OCR error is 11%.

An interesting framework for detection and tracking of scene text, although not for athletes identification, is proposed in [15], where particle filter is employed for robust text tracking, while text detection relies on connected components and texture analysis. The time performance of detection and tracking algorithm, reading not included, varies from 5 to 13 fps on different grey-level videos recorded at 640×480 resolution at 15 fps. Tracking results are available as on-line videos, but they are not provided in terms of precision and recall.

In [19] the textual regions to be tracked are assumed to be planar in the scene. To track a text region, possibly undergoing scale changes and 3d rigid motion, a sufficient number of points of interest is extracted. Small regions around them are tracked using normalized correlation in order to estimate the planar transform over a block of multiple frames and therefore to estimate accurately the motion of the text regions. However, this work applies to tracking off-line and is not suitable for real-time processing. Provided results of this text tracking algorithm are interesting but preliminary and the target to be tracked is manually initialized to simulate the results of a text detection process.

In Table 1 we summarize only the reviewed works which present a complete system including detection, identification and tracking of the athletes, and providing quantitative results. In Section 7 our method is compared with them.

Table 1: Performance of related works on athlete identification by reading text on the jersey.

method	precision	recall	data-set	text	figures at
[33]	84%	62%	200 frames	number	frame level
[33]	87%	77%	30 clips	number	frame level
[2]	83%	56%	36 shots	number	shot level
[3]	84%	68%	40 shots	number	shot level
[24]	87%	29%	1116 frames	number	frame level
[24]	71%	65%	+39 shots	number	shot level

3 Outline

In this section we present the architecture of our method for the automatic annotation of athletics video. The system aims to provide a list of visible athletes for each frame along with the approximate location inside the image. The diagram reported in Figure 2 depicts the three main blocks, along with their input/output and control flow. The input is a video sequence that is analysed on a frame-by-frame basis, along with some of a-priori knowledge about the context. As a final output, a list of xml encoded data is generated containing the annotation for each frame.

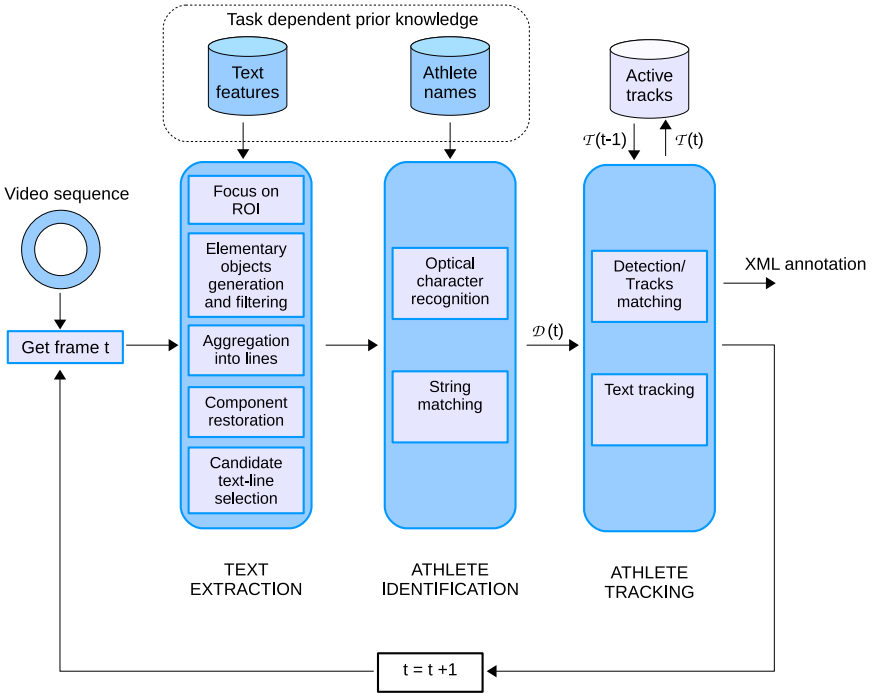

Figure 2: Outline of the presented method: Details of the blocks concerning the steps of text extraction are reported in Section 4. The blocks for athlete identification are described in Section 5. The method used for text tracking is explained in Section 6.

The *Text Extraction* block, described in Section 4, receives frames along

with parameters relating to specific knowledge about the appearance of the text we are interested in. A detailed description of the core method used for extracting text can be found in this previous work [16]; however, main adaptations aimed at exploiting available prior knowledge and the restoration step are detailed in this paper.

In the *Athlete Identification* block, illustrated in Section 5, each candidate text line is fed into an OCR which plays a double role: on one side it filters away false alarms detected as text, on the other side it provides a transcription of the input as a string of characters. The distance of this string from the known athletes' names provides a confidence measure for the presence of each athlete in a frame. The output of this module is a list of detections, i.e. athlete names with recognition scores and minimum bounding rectangle coordinates which indicate their location in the image.

The purpose of the *Athlete Tracking* block, described in Section 6, is to update the list of active Tracks (the list of athletes found in the current frame), by integrating the list of current detections with information from the previous frame. The current detections are compared to active tracks to either determine correspondences with existing tracks or to initiate new ones. Active tracks that do not have a correspondence in the current frame are potentially continued by searching for visually similar zones based on previous appearances. This last step permits the system to track athletes even when their names are visible yet unreadable, and typically provides a significantly denser labelling of the video sequence.

4 Text extraction

To extract the text in a single frame we apply a modified version of Messelodi and Modena's method [16] with some adaptations. The paradigm is that of connected component generation followed by component selection using attribute filtering, and a divisive hierarchical clustering method to produce candidate text lines.

In this new implementation, we exploit the knowledge about the context, in order to improve detection rate and time performance. To this purpose we introduce a pre-processing step which focuses the algorithm [16] on certain zones of the images. Furthermore, two novel post-processing steps have also been added to restore erroneously filtered objects, and to check candidate lines before the OCR step. The steps of text extraction are indicated in the

first block of Figure 2 and detailed in the following.

Focus on region of interest In order to reduce the risk of false alarms, to speed up the computation of the whole system, and to avoid passing to the OCR un-interesting strings, the computation is focused on a subset of the image. First of all, the upper part of the image is cropped using the common-sense constraint that a cameraman usually frames the image so that the athlete's bib is not in the upper part of the image (hence cutting through the athlete's face). The second prior knowledge that is used, is that of black-on-white text of this scenario; by exploiting this characteristic the attention is focused only on unsaturated colour zones in the image, in particular white zones. These zones are obtained by selecting all of the pixels that exhibit a relatively high luminance value L and a low saturation value S. Regions of interest are then obtained by discarding small areas (using an adaptive threshold which depends on the shortest athlete's name in the list), and considering the convex hulls of the remaining regions. We call the union of the resulting regions *background zones*. Figure 3(b,c) illustrates an example of selected pixels and the subsequently formed *background zones*, respectively. In this way the text detector [16] acts on a smaller number of pixels, depending upon scene content.

In scenarios where text foreground and background have polarities different from black-on-white, a different criterion must be applied, using a rough colour quantization algorithm in order to extract the pixels of the image having approximately the text background colour.

Generation of elementary objects The first step relies on an intensity normalization process to compensate for light variations throughout the image. Normalization is achieved by the computation of the divisive local contrast:

$$N(x, y) = I(x, y)/(A_w(x, y) + b) \tag{1}$$

where $I(x, y)$ is the intensity value of the pixel in (x, y), $A_w(x, y)$ is the average intensity computed inside a squared neighbourhood centred about (x, y), w is the dimension of the moving window, and b is a bias term enforced to avoid dividing by zero. It is analogous to unsharp masking replacing subtraction. $N(x, y)$ is computed only on pixels (x, y) belonging to the *background zones*, while the rest of N is set by default to 1. To speed up the computation

of N, we pre-compute the summed-area table [4] (also known as the integral image) for the local average values.

This operation improves image details and the local contrast in shadowed regions, agreeing with Wertheimer's contrast invariance principle: "image interpretation does not depend on actual values of the grey levels, but only on their relative values" [5]. This is particularly true in the case of text interpretation. In general, the optimal value of w depends on the scale - in pixels - of the structure to be detected. We fixed this size at an average value of the text thickness, using some examples of text which we expected to be successfully extracted and read. Thus, w can be regarded as a function of image resolution.

Two thresholds can be determined by taking into account the shape of the histogram of N:

$$t_1 = m - d_L/2 \quad (2)$$

$$t_2 = m - d_R/2 \quad (3)$$

where m is the histogram mode, and d_L, d_R are the left and right deviation from m, respectively. These are then used to extract two binary maps which should contain, respectively, positive and negative contrasting text. The connected components of these bitmaps constitute the *elementary objects*.

The domain knowledge contains a coded information about positive and/or negative contrast of the text of interest, driving the system to the analysis of the first, the second, or both the bitmaps. In our case, the prior knowledge suggests that only the first bitmap need to be addressed, working under the hypothesis that the athletes' names or numbers are darker than their surrounding background. An example of normalized image N (obtained with $w = 7$) computed on the *background regions* is reported in Figure 3(d). In this example, the left threshold, computed from the analysis of the histogram of N, was 0.97.

Filtering of objects In this step, features of the elementary objects are analysed by a cascade of attribute filters to mark likely non-text components as non-interesting. Each discarded component is marked with a label identifying why it was rejected, using the following filtering criteria:

- its area is very small (as readable text cannot be too small);

- it touches the image border (as a character lying near to the periphery is likely to be incomplete and thus not easily readable);
- its height is too short or too tall to be target text;
- its elongation, computed as the ratio of the equivalent ellipse's axes is high (only components whose area exceeds a certain threshold are considered in this test);
- its delimitation is not sharp, i.e. the percentage of border pixels exhibiting a sufficiently high gradient is too low.
- it is not significantly embedded inside the focus zone, i.e. a high percentage of outer boundary pixels do not belong to the *background regions*.

An example of elementary object labelling, is reported in Figure 3(e), where the survived elementary objects are in black, and those filtered away are labelled with a colour according to the filtering criterion.

Each filter requires us to devise one or more thresholds. Some parameters, such as area or size, as well as the normalization window dimension w, are related to the readability of extracted characters. The threshold levels depend upon the image resolution and are set to minimize the discarding of potentially good characters. Some thresholds are independent of resolution, as their corresponding features (e.g. elongation) are dimensionless, normalized quantities in the range of $[0, 1]$; in these cases reasonably slack thresholds are chosen.

Through experimentation, we found that the *delimitation* test is very important in order to filter out many non-relevant components captured by the thresholding operation. Text which is intended to be read at a distance, is usually created with a strong gradient with respect to the background, although this can be weakened by poor illumination conditions or blurring. According to the Helmholtz perception principle [5], for analysed components to be meaningful they should have an obvious perceptual boundary. In cases where an image is slightly out of focus, if we were to use a fixed threshold on delimitation we would risk to discard all text components as the border is not well defined, whilst the components per-se would remain quite readable. Therefore, the threshold on the gradient value used to compute the delimitation is adaptively estimated through the analysis of the image sharpness.

Its reference value is adaptively computed by averaging the gradient of the grey level values of the border pixels of all of the components in bitmap, thus exploiting scene statistics.

Aggregation into lines Next, surviving components are recursively clustered according to proximity, alignment and size similarity, until a termination criterion is satisfied. Clusters which potentially contain a single text line are extracted. One major difference with respect to [16], is that we have introduced a constraint in the alignment criterion in order to cluster (within an angular range from horizontal), thus exploiting prior knowledge about the target text type. The angular range is specified in the knowledge input file. Not all of the components are represented in the final clusters, thus only clusters that satisfy text line characteristics are considered.

In the example reported in Figure 3(f) four groups of elementary objects aggregated into lines are depicted with different colour.

Component restoration Once a cluster is accepted as a candidate textline, all of the components inside this region, which were previously marked as non-interesting, are reconsidered for inclusion. The idea is to recover characters, or fragments of characters, that were discarded in the filtering phase. For example, thin components, like the letter *I*. Their restoration is often useful to improve the input for the OCR module.

The restoration algorithm concentrates its attention inside the convex hull of the candidate line, considering all the elementary objects lying inside. Any new candidate components are checked again. Those smaller than a certain area are discarded, but in this case the considered area threshold θ'_A is lower than θ_A used in the global filtering step ($\theta'_A = \theta_A/4$). The delimitation index of the components is tested before being restored, decreasing the gradient reference value.

Candidate text line selection As a final step before attempting to read the text, a set of filters is applied in order to discard possibly spurious lines. Some examples of candidate text lines are depicted in Figure 4. By reducing the number of text lines, system processing time can be dramatically reduced as the OCR step is one of the most demanding in the whole chain. The tests depend upon a set of thresholds whose values rely on the prior knowledge or have been experimentally set through the analysis of samples of positive

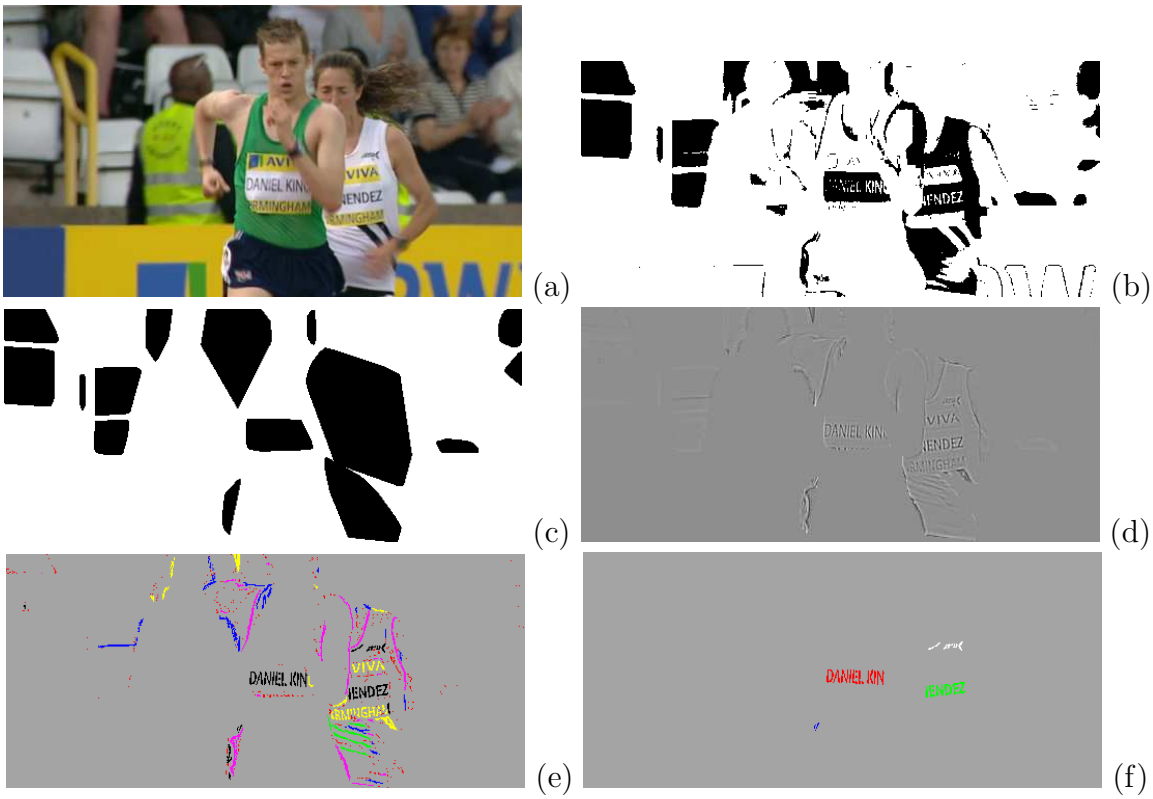

Figure 3: Text Extraction steps. (a) Input image. (b) Focus on region of interest: pixels with text background colour (here white), after image cropping. (c) Convex hull of the connected components which have a sufficient area to contain text characters; in this example the *background zones* cover 20% of the total pixels. (d) Intensity normalization map computed only in the region of interest, for the generation of elementary objects with adaptive binarization. (e) Elementary object labelling (here 752 in total), coloured according to different filtering criteria. Survived components are candidate to be text; here are marked in black (27 in total). (f) Candidate text lines (here 4) obtained by clustering by nearness and alignment of the survived elementary objects. In this example one cluster, the smallest one, will be filtered away by the last line selection procedure, before the OCR step.

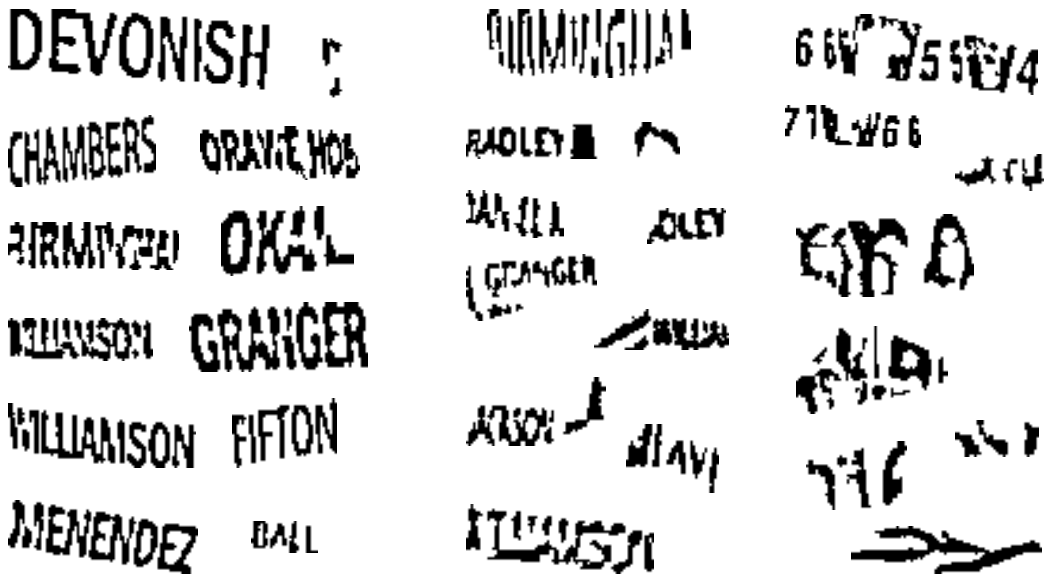

Figure 4: Examples of cluster before the candidate text line selection.

(text lines) and negative (non text lines) cases all extracted from a training set. The cascade of filters applied to lines are as follows:

- Possibly broken characters, or composite characters like "i" are aggregated into a single unit. Using the prior knowledge about the specific scenario, a filter is applied to the candidate lines to exclude those formed by too few units with respect to a fraction of characters building the shortest name in the list.

- Lines whose aspect ratio (width/height of the minimum bounding rectangle) is out of a range are discarded. The range is computed from the aspect ratio of a fraction of the shortest name and a factor of longest name in the list, as would typed in a standard sans-serif font.

- The gradient inside a candidate line region is computed and if the average gradient magnitude is below a threshold (4% of the highest possible gradient value in our experiments) the line is discarded.

- Colours inside the line zone are tested, which should be concentrated on the text and background colour. In our black-on-white case, we classify the pixels into three classes: *coloured* if the pixel saturation

is above a threshold T_{sat}, *white* if the pixel is not coloured and its luminance is above a threshold T_{lum}, *black* otherwise. A candidate line passes through the filter if the percentage of *coloured* pixels is below a threshold T_{col} and the percentage of *white* pixels is above a threshold T_{white}. In our experiments we use the following values: $T_{sat} = 40$, $T_{lum} = 128$, $Th_{col} = 10\%$, $T_{white} = 50\%$. System performance appears to be robust with respect to small variations of these values.

Clusters which survive are subsequently passed on to the character recognizer for classification.

5 Athlete identification

An athlete's identity can most robustly be established through two visual modalities: face recognition and name/number reading. In this paper, we only address the second one, whilst fusion with the result of a face recognizer is left to future work.

5.1 Text reading

The output supplied by the text extractor is composed of a set of candidate text lines, i.e. clusters made up of connected components of binary pixels. To interpret them, we apply an OCR system which transcribes the pictorial representation of text into a coded version, where a unique code (e.g. ascii, utf-8) is assigned to each different character. The OCR system selected for our experiments is *Tesseract* [28], a free optical character recognition engine originally developed as proprietary software by Hewlett-Packard between 1985 and 1995. Hewlett-Packard released it as open source in 2005 and it is now available at `http://code.google.com/p/tesseract-ocr` under the Apache License 2.0. Tesseract proved itself be one of the top three OCR engines in the 1995 Annual Test of OCR Accuracy [23] and it is now considered to be one of the most accurate free OCR engines currently available. Tesseract 2.04 is a pure OCR engine, and does not include document layout analysis and output formatting. Since our text detection/extraction module produces a list of separated text lines (or words), Tesseract is ideal for our recognition purposes. We also explored the possibility of training Tesseract for specific fonts using sample characters extracted from test images, but found that the

"eng" language, provided within the standard distribution, is adequate for our task. We used the list of athletes' names as its "user-words" file.

After running the text extractor, the connected components grouped into candidate words are converted into a raster image, for subsequent feeding into the OCR system. The output of the OCR is a string of characters accompanied by an index, in the range $[0 - 100]$, quantifying an average recognition confidence. Let us consider, as an example, the processing of the frame in Figure 5 (left), where the detected candidate lines are superimposed (right). The text detector has located three candidates, two of which are actual text and are recognized as "DANIEL KIN" (OCR confidence 83) and "IENDEZ" (confidence 69). The third one is a false alarm which is interpreted by the OCR as "MMM" (confidence 15). We can filter away lines with a low OCR confidence, as they are likely to be derived from noise.

Figure 5: OCR results for candidate text lines. The red line is read as "DANIEL KIN" with OCR confidence 83, the white one as "MMM" with OCR confidence 15, and the green one "IENDEZ" with OCR confidence 69. The OCR confidences here act as a filter on the candidate athletes' names.

5.2 Distance from athlete names

The results delivered by the OCR are far from accurate for several reasons: low text resolution, poor quality image resulting from compression artefacts, non-textual components erroneously passed as an input, distortion of character shapes, false character connections, text occlusions, fragments lost, etc. Nevertheless, the output often appears to be significantly similar to one of the actual names we know to be taking part in the event, thus making it feasible to identify an athlete's name by measuring the similarity between the OCR output string and each of the candidate names. The edit distance

proposed in [17] has been employed to compare each athlete's name with the OCR results. This check returns a value between $[0, 1]$, where 0 represents complete dissimilarity, and 1 indicates a perfect match between the strings. We consider a detection to be valid only if an athlete has a high similarity score with the OCR-ed text. In the case where two (or more) strings from the same frame deliver the same name, only that which has a higher similarity is kept.

The output for the frame t is a list of detections $\mathcal{D}(t)$ indicating the athletes identified. Each element $D_{i,t}$ of $\mathcal{D}(t)$ is a 4-tuple (N_i, B_i, S_i, V_i) constructed from: the athlete's name N_i, the coordinates of the bib position B_i (namely the bounding box), the string similarity value S_i and a visual signature V_i. The latter is a quantized colour histogram of the region cropped by B_i in the frame t.

6 Athlete tracking

The identification of athletes based solely on the detection and reading of text on their uniforms (especially when coupled with a knowledge about how text should appear and a list of expected names) typically provides a highly reliable output. Unfortunately, due to a variety of reasons, only on a small subset of video frames (i.e. when the text is readable), will provide such a text identification output. When considering sports applications, two critical factors affect the readability of text: (i) athletes move in a non-rigid way, quickly changing postures; (ii) cameras are non-static as pan/tilt/zoom operations are applied in order to better follow the event. Motion blur, partial occlusions and bib deformations also complicate or make it impossible to read text. As a direct consequence, any video annotation system based solely on the reading of text on a frame-by-frame basis can only provide sparse annotations.

To improve the denseness of the annotations, temporal continuity can be exploited, trying to best fill gaps in between successive detections, taking into consideration other features typically based on the visual appearance of the athlete's bib.

The text tracking module follows athletes by updating an *active tracks* list $\mathcal{T}(t)$ for each frame t. Each element $T_{j,t}$ of $\mathcal{T}(t)$ is a 4-tuple (N_j, B_j, S_j, V_j) constructed from: the athlete's name N_j, the bounding box coordinates of the bib position B_j, a confidence score of the track S_j and a visual signature

V_j (a quantized colour histogram inherited from a detection).

$\mathcal{T}(t)$ is populated on a frame-by-frame basis taking into account the output of the athlete identification step $\mathcal{D}(t)$ and the active list $\mathcal{T}(t-1)$. The active list $\mathcal{T}(t)$, is computed in two steps, by comparing the detections of the frame t with the active tracks of the frame $t-1$, and by tracking only those having no correspondence in the detections set.

First step - Comparing current detections to active tracks The lists $\mathcal{D}(t) = \{D_i, i = 0 \dots n\}$ and $\mathcal{T}(t-1) = \{T_j, j = 0 \dots m\}$ are compared in order to find name and/or location correspondences. We divide the possibilities into five groups:

1. *full correspondences*: pairs $\{D_i, T_j\}$ relating to the same athlete, i.e. $N_{i,t} = N_{j,t-1}$, and whose bounding boxes $B_{i,t}$ and $B_{j,t-1}$ match;

2. *name correspondences*: pairs $\{D_i, T_j\}$ such that $N_{i,t} = N_{j,t-1}$, but boxes do not match;

3. *location correspondences*: pairs $\{D_i, T_j\}$ such that the bounding boxes $B_{i,t}$ and $B_{j,t-1}$ match, but names are different;

4. detections that do not correspond to any T_j, $j = 0, \dots m$;

5. active tracks that do not correspond to any D_i, $i = 0, \dots n$.

The match between bounding boxes is defined in term of their spatial overlap: they match if the ratio between the intersection area and the union area overcomes a prefixed threshold. In this case, we make the reasonable assumption that the variation of bib location in two successive frames is limited.

1. For each full correspondence $(D_{i,t}, T_{j,t-1})$, the detection $D_{i,t}$ is used to define a track to be inserted into $\mathcal{T}(t)$: the new track is $(N_{i,t}, B_{i,t}, \max(S_{i,t}, S_{j,t-1}), V_{i,t})$.

2. Concerning the name correspondences $(D_{i,t}, T_{j,t-1})$, the decision about keeping the information contained in $D_{i,t}$ is postponed to the end of the tracking step described below, i.e. considering the result of tracking $T_{j,t-1}$ in frame t.

3. If $(D_{i,t}, T_{j,t-1})$ is a location correspondence, the decision about keeping information concerning $D_{i,t}$ depends upon the comparison of the detection score $S_{i,t}$ and the tracking score $S_{j,t-1}$. If $S_{i,t} \geq S_{j,t-1}$, the new track $(N_{i,t}, B_{i,t}, S_{i,t}, V_{i,t})$ is inserted into $\mathcal{T}(t)$, otherwise $D_{i,t}$ is erased and $T_{j,t-1}$ is moved into the group of tracks without correspondence (fifth group).

4. For each detection $D_{i,t}$ without correspondence the new track $(N_{i,t}, B_{i,t}, S_{i,t}, V_{i,t})$ is added to $\mathcal{T}(t)$.

5. For each active track $T_{j,t-1}$ from the last group, the second step is executed.

Second step - Tracking active tracks In this step, only active tracks $T_{j,t-1}$ from the second and the fifth groups are involved. The procedure consists of searching in the current frame t for a sub-image similar to that of the bib region defined by $B_{j,t-1}$, investigating the neighbourhood around the previous location, using template matching.

Let R be the bib region extracted from the previous frame $t-1$ using $B_{j,t-1}$ whose starting point is (x_B, y_B). From this point, the algorithm repositions R in the current frame t inside a restricted region around the point (x_B, y_B). For each position of R, we compute a similarity measure based on the L^1 distance between R and the corresponding region. In order to reduce computational cost, we apply a standard multi-resolution technique through the creation of two pyramids, for the template R and for the region of interest in the current frame. To achieve this, images are first convolved with a blurring filter and then subsampled to get their lower resolution versions. A best-match search proceeds from coarse to fine resolutions in the following manner: the smallest template is matched against the smallest image, with the location of the minimum distance being identified; matches in subsequent pyramid levels are limited around the corresponding location. The result of the template matching step is a tracking score M_s and a tracking position M_p, defined respectively as the minimum L^1 distance and the corresponding location in the current frame.

After tracking we check two conditions on the score M_s and on the visual similarity between the track $T_{j,t-1}$, stored in $V_{j,t-1}$, and the appearance extracted from the current frame t in the position M_p. If both (i) M_s is below a given threshold and (ii) the two appearance histograms are close

enough (i.e. their L^1 distance is below a threshold), then the new track $(N_{j,t-1}, M_p, S_{j,t-1} \times f(1 - M_s), V_{j,t-1})$ is inserted into $\mathcal{T}(t)$. The score of the new track is modulated by multiplying the previous track score $S_{j,t-1}$ by a function f of $(1 - M_s)$, in order to reward tracks characterized by good bib-region similarity throughout the frames. The second condition has been introduced in order to mitigate the typical problem known as track-drift, a phenomenon in which the target gradually shifts away from its original appearance.

Otherwise, i.e. at least one of the two conditions is not satisfied, if $T_{j,t-1}$ belong to the second group the information from $D_{i,t}$ is retained: the new track $(N_{i,t}, B_{i,t}, S_{i,t}, V_{i,t})$ is inserted into $\mathcal{T}(t)$.

As an example of tracking let us consider the case represented in Figure 6. In the start list of the event there are, among others, two athletes identified by numbers 429 and 438. At frame t (the first one in the Figure) a text detection occurs which wrongly identifies the athlete as 429 with score 0.857 (OCR output is the string 4129). It falls in the group 4: it is inserted as a track into $\mathcal{T}(t)$. At frame *t+1* no detection occurs and the tracking step is applied to the track in $\mathcal{T}(t)$ (group 5): tracking score is good (0.951) as well as visual similarity. Hence the track is inserted into $\mathcal{T}(t+1)$ with the modulated score 0.846.

At frame *t+2* athlete 438 has been correctly detected (with score 0.666), in a position overlapping the track in $\mathcal{T}(t+1)$: it is a location correspondence (group 3) where the score of the track (0.846) is greater than the score of the detection. The detection is ignored and the track is regarded as a track without correspondence (group 5). The tracking is performed: the tracking score is 0.953 and the visual signatures are similar. Hence the track is inserted into $\mathcal{T}(t+2)$ with score 0.835. At frame *t+3* athlete 438 is correctly detected with score 1.0 in a location which overlaps the track in $\mathcal{T}(t+2)$. It is a location correspondence (group 3) where the score of the detection (athlete 438) is greater than the track score (athlete 429). Then the detection is inserted into $\mathcal{T}(t+3)$ with its score.

Once $\mathcal{T}(t)$ has been computed, an *xml* encoded output is generated to provide a stream of information concerning athletes' presence at frame t and their corresponding position inside the image.

Figure 6: Four consecutive frames of a sequence to illustrate an example of the tracking procedure. Bounding boxes superimposed represent the location of the active tracks.

7 Experimental results

Considering the aim of the module inside the My eDirector 2012 project, we extensively tested the method on undirected athletic video sequences. Furthermore, to compare properly our performance with those of other systems available in the literature, we tested the method on several edited videoclips taken from the Internet.

We evaluated the performance through the comparison of the identity/position of the athletes proposed by the system to their actual position in the frames. Although manually counting correct outputs is a tedious task, a user can be provided with an interface in which the athlete's name is superimposed over the frame in the found location, thus aiding a visual verification of correct and false detections and identifications. Counting missed detections is not so simple, however, as it requires the user to establish whether the name of the athlete is visible and readable in the scene. Such a task poses several difficulties, especially in cases of borderline readability, cropped text, partially occluded text, blurred or small text.

To provide an estimate of the real occurrences we need to visually check every single frame and note the athletes which we expect the system should be able to identify solely through text. Of course this is a rather vague definition. Qualitatively, we decided to take into account only athletes for which (i) the bib is un-occluded, or almost completely visible, (ii) the bib presents an approximately frontal view, and (iii) the text has a resolution of at least 6 pixels per inch. These criteria can only provide a rough estimate of actual occurrences, however, we believe that it can sufficiently deliver an idea of the system's limits and recall performance. As an example, we can consider the frames in Figure 7: in the first row (from left to right) we have two athletes matching the above criteria, then one, and finally three. But, images in the second row do not contain any views of valid athlete to be

inserted in the ground-truth.

Figure 7: Examples of frames where the athletes are present at different resolutions: close shot, close medium shot, long shot, far field, very far field, no athlete.

We have included in the ground-truth occurrences of athletes framed in long shots, although the text often has limited readability. In such scenes possibly only the tracking of an already identified athlete, but not the text reader, can produce a correct labelling. As a consequence, the relative frequency of athletes framed in close, close medium, and long shots in a sequence heavily affects the recall rate.

7.1 Undirected video sequences

We have extensively tested the presented method on eight undirected video sequences taken from a large data-set created for the My eDirector 2012 EU project. Five sequences, *walk-cam02, walk-cam06, longjump-w-cam12, longjump-w-cam13, longjump-w-cam14*, have been recorded for BBC in the Birmingham athletics stadium in 2009, and three sequences (*longjump-m-ch05, longjump-m-ch06, longjump-m-ch07*) have been recorded for BBC in the Crystal Palace London athletics stadium in 2010. The first two sequences last 24' 32" and were recorded by two cameras observing the 5000m walk. The remaining three Birmingham videos last 15' 42" and relate to a long-jump women event observed by three cameras from different point of views. Finally, the three London videos last 20' each one, all concerning a long-jump

men event. For all the sequences the frame rate is 25 fps, the frame size is 1024×576 pixels and the text dimensions of the name depend heavily upon the framing: ranging from close shot to very far field. Framing examples are reported in Figure 7).

Sequences *walk-cam02* and *walk-cam06* cover the same event, however, one always offers a closer view with respect to the other. We estimate that about three quarters of *walk-cam06* sequence depicts athletes in far field, less than 10% are close and close medium shots, and the rest are long shots with borderline readability. In both the sequences two or more athletes are often depicted in a frame. The same happens for the long-jump women event. The *longjump-w-cam14* sequence frames the athletes closer with respect to *longjump-w-cam12* and *longjump-w-cam13*. In *longjump-w-cam12*, athletes are captured during the running and jumping phases, in *longjump-w-cam13* they are captured by a camera that is located just after the landing zone. Finally, sequence *longjump-w-cam14* depicts a close medium shot of the athletes while they are preparing to jump. Concerning the long-jump men event, cameras are devoted to different phases of the jump, with infrequent field overlap. In *longjump-m-ch05* the athletes are always quite far. In the other two sequences athletes are framed from close shot to long shot; a considerable percentage of the sequence *longjump-m-ch07* is garbage footage, framing the ground.

The output generated by the proposed module for the eight test videos is generalised in Table 2. In this table, the comparison between system output and ground-truth is reported: the number of labelled regions generated by the system (*system output*), the number of correct athlete localizations and identifications (*correct id*), the number of errors due to incorrect athlete identifications (*false id*) and to incorrect localizations (*false loc*).

Performance of the system, in terms of precision and recall, are reported in Table 3.

The precision rate ranges from 88% to 100% (on average is around 98.9%). The relatively low precision of *walk-cam06* is due to a single false localization tracked for more than 6 seconds. For the sequence *longjump-m-ch05* neither precision nor recall are meaningful: The athletes are framed in far field, so their recognition through text reading is not practicable. We have included this sequence to test deeply the method on false alarm rate. The 90 false localizations correspond to only one false detection tracked for less than 4 seconds.

The recall rate ranges from 12.9% to 89.2% (on average is around 45%).

Table 2: Athlete identification results for the eight undirected sequences.

sequence	frames	ground truth	system output	correct id	false id	false loc
walk-cam02	36800	27 500	9814	9715	80	19
walk-cam06	36800	10 360	1514	1333	8	173
longjump-w-cam12	23550	10 600	2192	2192	0	0
longjump-w-cam13	23550	5 400	1784	1777	6	1
longjump-w-cam14	23550	11 900	10614	10613	0	1
longjump-m-ch05	30000	0	90	0	0	90
longjump-m-ch06	30000	14 080	9399	9374	11	14
longjump-m-ch07	30000	5 630	3484	3460	0	24
TOTAL	234250	85 470	38891	38464	105	322

In the *walk-cam02* sequence the recall rate is 35%, whilst in *walk-cam06*, which generally covers the event in long and far-field, the estimated recall rate drops to 13%.

In *longjump-w-cam12*, athletes are captured during the running and jumping phases using a wide field-of-view perspective, consequently the text is often blurred and its resolution is at the lower limit of the acceptable range. Here, the recall rate is about 20%. In the *longjump-w-cam13* sequence, athletes are running towards the camera and after the jump they are very close, thus framed from the waist upwards, making the text readable in many cases. The recall rate for this camera is at 33%. Sequence *longjump-w-cam14* depicts athletes while they are preparing to jump with medium shots. In this condition the text is often sharp and large, and the recall rate moves up to 89%. The comparison of the system output for the last three sequences covering the same long jump event is shown in Figure 8. Considering that the identification is easier in close and close medium shots, this graphical representation puts into evidence the different framing of the cameras.

Finally, for *longjump-m-ch06* and *longjump-m-ch07*, where long shots are not the major part, recall is 66.6% and 61.5%, respectively.

The recall figures vary in a wide range putting in evidence the limits of the algorithm in long shots. Nevertheless, it is important to note that the proposed module provides a reliable output throughout close and close medium sequences.

Table 3: Athlete identification performance for the eight undirected sequences.

	precision	recall
walk-cam02	99.0%	35.3%
walk-cam06	88.0%	12.9%
longjump-w-cam12	100.0%	20.7%
longjump-w-cam13	99.6%	32.9%
longjump-w-cam14	100.0%	89.2%
longjump-m-ch05	-	-
longjump-m-ch06	99.7%	66.6%
longjump-m-ch07	99.3%	61.5%
TOTAL	98.9%	45.0%

The processing time (reported in Table 4), has been calculated on a Linux platform endowed with a 2.83GHz CPU and 4 GB RAM. The table reports the average processing time per frame for the five Birmingham sequences, for the three main blocks of the system. We can observe that the processing rate is adequate for a 25 fps video feed, thus real-time processing is possible. While text extraction depends on the complexity of the scene, athlete identification depends on how many candidate lines are processed by the OCR.

Table 4: System processing time (in milliseconds). For each Birmingham sequence, the average time to process a frame is reported (last row), along with the average time for the three main modules of the system.

time (msecs)	*cam02*	*cam06*	*cam12*	*cam13*	*cam14*
text extraction	15.38	12.89	17.56	18.22	14.87
athlete identification	13.16	7.15	3.43	6.42	5.59
athlete tracking	5.61	0.28	0.83	1.14	4.14
total (per frame)	34.15	20.33	21.82	25.79	24.60

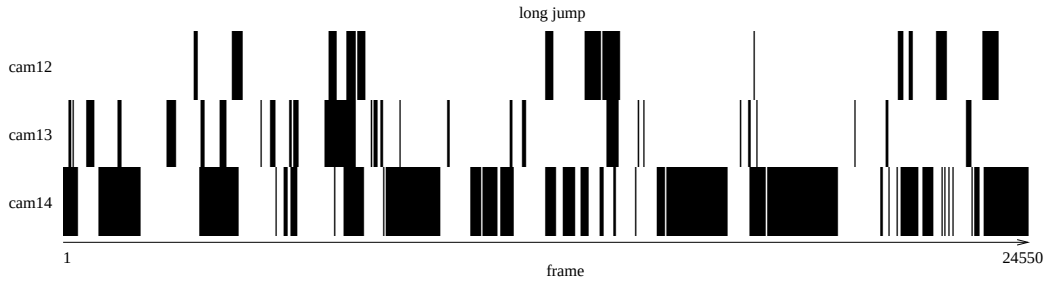

Figure 8: Athlete detection through text reading/tracking in the video sequences *longjump-w-cam12*, *longjump-w-cam13* and *longjump-w-cam14*. The identification is more frequent in the frames of *longjump-w-cam14* because this camera zooms in on the athletes as they prepare to jump.

7.2 Directed videoclips

Although our method has been developed to work on undirected videos (electronic direction is one of the aim of the My eDirector 2012 project), supplementary experiments have been performed on directed videoclips, mainly for a comparison with the state of the art.

We collected from Internet several edited high resolution videoclips concerning athletics Youth Games of Singapore 2010, where athletes are identified by a three digit numbers on the bib, instead of the name (Figure 6). The duration of the videoclips ranges from one to two minutes, and the frame size is 1280×720. This footage is rather heterogeneous presenting different athletic games under various illumination conditions (natural, artificial by night, raining).

The format of the videoclips, edited with different kind of transitions, may include a panoramic view of the stadium, the superimposed start list, close-up shots of some of the athletes with caption, the live event, the replay, the superimposed result table and the victory ceremony.

Results are summarized in Table 5 where the *ground-truth* column reports the manual count of bib views in the frames, *system output* is the number of identifications proposed by the system, *false id/loc* are the errors due to false identification or to false detections, *precision* is the percentage of correct identifications on system outputs, *recall* is the percentage of correct identifications on expected output.

Edited videoclips present some characteristics which pose further difficulties for our method. Some situations, that are considered anomalous in

Table 5: Results of our athlete identifier on directed videoclips.

sequence	frames	ground truth	system output	correct id	false id/loc	precision	recall
100m W	2 378	801	901	792	109	88.0%	98.9%
100m M	3 001	1 178	1 167	965	202	82.7%	81.9%
200m W	1 797	520	341	332	9	97.4%	63.3%
200m M	1 777	658	507	485	22	95.7%	73.7%
110mH M	1 757	796	451	451	0	100.0%	56.7%
Hammer W	2 304	347	212	204	8	96.2%	58.8%
TripleJ M	1 907	782	917	761	156	83.0%	97.3%
TOTAL	14 921	5 082	4 496	3 990	506	88.7%	79.0%

undirected videos, are frequent in the edited ones, like two athletes depicted in the same location in two subsequent frames, due to a cut, or an athlete in two different locations in the same frame due to a dissolve. Superimposed captions, chronometer, start and result list are not present in rough videos and some false alarms occur just in correspondence of them, when superimposed text is similar to an athlete identification number (e.g. in *100m W* the athlete with number 117 is erroneously identified in correspondence of the superimposed time 11.73).

We note a significantly different precision rate in the two sets of experiments (98.9% vs. 88.7%). The main reason is that it is generally easier to identify athletes by name than by short numbers, which can be confused not only with overlaid text but also with other numbers often present in the scene.

The difference in the recall figures (45.0% vs. 79.0%) depends on the type of videos: undirected vs. directed. As mentioned, the frequency of close, close medium, and long shots included in the ground-truth influences the recall rate. In edited videoclips, the selected shots present a higher percentage of athlete close-ups. For example, the percentage of athlete occurrences appearing in close and medium shots, is almost 100% in *100m W*, 66% in *200m W*, 64% in *110m Hurdles M*, while the percentage is about 15% in *walk-cam06* and about 30% in *longjump-w-cam12*.

Experiments on videoclips where athletes are identified by numbers per-

mit a meaningful comparison with the state of the art works summarized in Table 1. We computed performance using the frame or the shot as basic unit, taking into account that in our videoclips more than one target can be present in the same unit.

The results reported in [33] for detection and tracking in videoclips, are 86.9% and 76.7% for precision and recall, respectively, provided at frame level. Correspondent figures of our method are 88.7% and 79.0%.

To compare our method with results reported by [3] and [24], we manually split the videoclips into shots in correspondence of transitions (cuts, dissolves) and annotate each of them with the athletes, from the start list, who are present in the shot. The athlete is considered present if she is identifiable by her number in at least 10% of the frames of the shot. Data set and results using the shot as validation unit are reported in Table 6.

Table 6: Results of our athlete identifier on directed videoclips computed at shot level.

sequence	shots	shot with athletes	nr of athletes	system output	correct id	false id	miss
100m W	18	8	8	8	8	0	0
100m M	19	5	7	7	7	0	0
200m W	12	5	7	5	4	1	3
200m M	13	7	8	7	7	0	1
110mH M	16	6	6	5	5	0	1
Hammer W	9	2	2	2	2	0	0
TripleJ M	7	3	5	7	5	2	0
TOTAL	94	36	43	41	38	3	5

The global precision and recall, at shot level, are 93% and 88%, respectively, while the correspondent figures in [3] are 84% and 68%. To properly compare the performance with the figures in [24], where precision is 71% and recall is 65%, we consider the output of our system on all the 94 shots, framing or not the athletes. Precision drops to 84% because the system output contains four more false alarms, while recall is obvioulsy the same. Our system remarkably out-performs both the methods in both the figures.

The comparison of our method with some of the related work presented in Table 1 is summarized in Table 7.

Table 7: Comparison with related works.

method	precision	recall	note
[33]	86.9%	76.7%	frame level
our	88.7%	79.0%	
[3]	84%	68%	shots with athletes
our	93%	88%	
[24]	71%	65%	all the shots
our	84%	88%	

8 Conclusions and future works

In this paper, a knowledge-based identification system for athletes in videos has been presented. A module that automatically detects and extracts embedded text from images is applied to each and every frame. Then, each extracted text region is passed to an OCR, which in turn provides a string representation for it. A comparison between these strings and the list of athlete names is then performed in order to generate a probability score that an athlete is portrayed in the image. When an athlete is identified with a sufficiently high confidence, the text zone is subsequently tracked by pyramidal template matching until a new text detection with a similarly high confidence occurs or it is no more trackable. Text detections with lower confidences (when available), are utilised to enforce tracking hypotheses in cases of low matching score. The text tracking strategy presented preforms well when matching blurred (unreadable) text, but it needs to be improved to compensate for scale changes of the target, which often occur due to the relative movements of athletes and cameraman zooming. Partial occlusions of the bib (due to an athlete's arm or a second athlete) are also frequent, therefore partial matching with the template should also be considered.

The precision, i.e. the probability that the output of the proposed method is correct, has been shown to be very good (98.9% in the performed experiments on undirected videos and 88.7% on directed videos). The recall rate computation requires a manual labelling of each athlete in each frame, to indicate if the text on their bib is visible in the scene (or not). Considering that visibility is difficult to define and that a huge effort is required to label long sequences at high frame rates, we have provided only a rough estimation

of the recall performance of the system, which is around 45% for undirected videos and 79% for directed videos, both computed on the presence of athletes at frame level.

A more reliable measure of recall in directed videoclips, 88%, is obtained considering the presence of the athlete at shot level, in this case precision is 93%. These figures improve in a significant way those of previous works in the literature.

The high precision of the system to localize and identify athletes suggests that it could be used as a trigger to initiate new tracks in a more general tracking system [21] (one which integrates several other visual features like face detection, face recognition, body appearance colour, skin detection, etc.), therefore providing a denser labelling of the video. As it is evident by the recall figures, missed detections in long shots is one of the main problem that will be investigated in the future work.

The whole system uses some simple but efficient techniques and experimental results on various data-sets show its high precision and fast speed. In fact, the processing rate is adequate for real-time processing of 25 fps videos.

Acknowledgments This work has been supported by the European Union under the Strep Project FP7 215248: My eDirector 2012. The authors would like to thank Paul Chippendale for his careful reading of the manuscript.

References

[1] E.L. Andrade, E. Khan, J.C. Woods, and M. Ghanbari. Player Identification in Interactive Sport Scenes Using Region Space Analysis Prior Information and Number Recognition. In *International Conference on Visual Information Engineering*, pages 57–60, July 2003.

[2] M. Bertini, A. Del Bimbo, and W. Nunziati. Player identification in soccer videos. In *7th ACM SIGMM International Workshop on Multimedia Information Retrieval*, pages 25–32, 2005.

[3] M. Bertini, A. Del Bimbo, and W. Nunziati. Matching Faces with Textual Cues in Soccer Videos. In *International Conference on Multimedia and Expo*, pages 537–540, July 2006.

[4] F.C. Crow. Summed-Area Tables for Texture Mapping. *Computer Graphics*, 18(3):207–212, July 1984.

[5] A. Desolneux, L. Moisan, and J.-M. Morel. *From Gestalt Theory to Image Analysis: A Probabilistic Approach*. Springer, New York, 2008.

[6] M. Mirmehdi (Ed.). Special Issue on Camera-Based Text and Document Recognition. *International Journal on Document Analysis and Recognition*, 7(2-3), July 2005.

[7] N. Ezaki, M. Bulacu, and L. Schomaker. Text Detection from Natural Scene Images: Towards a System for Visually Impaired Persons. In *International Conference on Image Processing*, volume 2, pages 683–686, August 2004.

[8] W. Jia, X. He, and M. Piccardi. Automatic License Plate Recognition: A Review. In *International Conference on Imaging Science, Systems and Technology*, pages 43–48, Las Vegas, Nevada, 2004.

[9] K. Jung, K.I. Kim, and A.K. Jain. Text Information Extraction in Images and Video: a Survey. *Pattern Recognition*, 37(5):977–997, May 2004.

[10] A. Kokaram, N. Rea, R. Dahyot, M. Tekalp, P. Bouthemyand P. Gros, and I. Sezan. Browsing Sports Video. *IEEE Signal Processing Magazine*, 23(2):47–58, March 2006.

[11] H. Li, D. Doermann, and O. Kia. Automatic Text Detection and Tracking in Digital Video. *"IEEE Transactions on Image Processing - Special Issue on Image and Video Processing for Digital Libraries"*, 9(1):147–156, January 2000.

[12] J. Liang, D. Doermann, and H. Li. Camera-based Analysis of Text and Documents : A Survey. *International Journal on Document Analysis and Recognition*, 7(2-3):84–104, 2005.

[13] R. Lienhart. Video OCR : A survey and practitioner's guide. In *Video Mining*, page Chapter 6, 2003.

[14] C. Mancas-Thillou and B. Gosselin. Natural Scene Text Understanding. In *Vision Systems: Segmentation and Pattern Recognition*, page Chapter 16, Vienna, Austria, June 2007.

[15] C. Merino and Mirmehdi M. A framework towards realtime detection and tracking of text. In *2nd International Workshop on Camera-Based Document Analysis and Recognition*, pages 10–17, 2007.

[16] S. Messelodi and C.M. Modena. Automatic Identification and Skew Estimation of Text Lines in Real Scene Images. *Pattern Recognition*, 32:791–810, 1999.

[17] E.W. Myers. An O(ND) Difference Algorithm and its Variations. *Algorithmica*, 1(2):251–266, 1986.

[18] G.K. Myers, R. Bolles, Q.-T. Luong, J. Herson, and H. Aradhye. Rectification and Recognition of Text in 3-D Scenes. *International Journal on Document Analysis and Recognition*, 7(4):147–158, 2005.

[19] G.K. Myers and B. Burns. A Robust Method for Tracking Scene Text in Video. In *1st International Workshop Camera-Based Document Analysis and Recognition*, pages 30–35, Seoul, Korea, August 2005.

[20] C. Patrikakis, A. Pnevmatikakis, P. Chippendale, M. Nunes, R. Santos Cruz, S. Poslad, W. Zhenchen, N. Papaoulakis, and P. Papageorgiou. Direct your personal coverage of large athletic events. *IEEE MultiMedia*, 2011.

[21] A. Pnevmatikakis, N. Katsarakis, P. Chippendale, C. Andreatta, S. Messelodi, C.M. Modena, and F. Tobia. Tracking for Context Extraction in Athletic Events. In *International Workshop on Social, Adaptive and Personalized Multimedia Interaction and Access - SAPMIA, ACM MM 2010*, 2010.

[22] EU FP7 Project. Real-Time Context-Aware and Personalized Media Streaming Environments for Large Scale Broadcasting Applications. http://www.myedirector2012.eu, 2011. [On-line; accessed 24 June 2011].

[23] S.V. Rice, F.R. Jenkins, and T.A. Nartker. The Fourth Annual Test of OCR Accuracy. Technical Report TR-95-03, Information Science Research Institute, University of Nevada, Las Vegas, July 1995.

[24] M. Saric, H. Dujmic, V. Papic, N. Rozic, and J. Radic. Player Number Recognition in Soccer Video using Internal Contours and Temporal

Redundancy. In *10th WSEAS international conference on Automation and information*, pages 175–180, 2009.

[25] T. Sato, T. Kanade, E.K. Hughes, M.A. Smith, and S. Satoh. Video OCR: indexing digital news libraries by recognition of superimposed captions. *Journal Multimedia Systems*, 7(5):385–395, 1999.

[26] S. Satoh, Y. Nakamura, and T. Kanade. Name-It: Naming and Detecting Faces in News Videos. *IEEE Multimedia*, 6(1):22–35, 1999.

[27] H. Shen and J. Coughlan. Finding Text in Natural Scenes by Figure-Ground Segmentation. In *International Conference on Pattern Recognition*, Hong Kong, August 2006.

[28] R. Smith. An Overview of the Tesseract OCR Engine. In *9th International Conference on Document Analysis and Recognition, ICDAR*, pages 629–633, Washington, DC, USA, 2007.

[29] P. Viola and M. Jones. Rapid Object Detection Using a Boosted Cascade of Simple Features. In *International Conference on Computer Vision and Pattern Recognition*, pages 511–518, 2001.

[30] J.J. Weinman, E. Learned-Miller, and A.R. Hanson. Scene Text Recognition Using Similarity and a Lexicon with Sparse Belief Propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(10):1733–1746, February 2009.

[31] W. Wu, X. Chen, and J. Yang. Detection of Text on Road Signs from Video. *IEEE Transactions on Intelligent Transportation Systems*, 6(4):378–390, 2005.

[32] J. Yang, M.-Y. Chen, and A. Hauptmann. Finding Person X: Correlating Names with Visual Appearances. In *International Conference on Image and Video Retrieval*, Dublin, Ireland, July 2004.

[33] Q. Ye, Q. Huang, S. Jiang, Y. Liu, and W. Gao. Jersey Number Detection in Sports Video for Athlete Identification. In *Visual Communications and Image Processing*, volume SPIE 5960, pages 1599–1606, Beijing, China, July 2005.

[34] J. Zhang and R. Kasturi. Extraction of Text Objects in Video Documents: Recent Progress. In *8th IAPR Workshop on Document Analysis Systems*, pages 5–17, Nara, Japan, September 2008.