

Distributed video acquisition and annotation for sport-event summarization

C. De Vleeschouwer¹, Fan Chen¹, D. Delannay¹, C. Parisot, C. Chaudy², E. Martrou³, and A. Cavallaro⁴

¹UCL/TELE, Louvain-la-Neuve, Belgium; ²ACIC sa, Mons, Belgium, ³Mediapro, ⁴QMUL, London, United Kingdom.

E-mail: christophe.devleeschouwer@uclouvain.be

Abstract: This document presents the video data set that has been recently collected within the FP7 APIDIS project [1]. The acquisition setting was deployed around a basket-ball court, and consists in a set of 7 calibrated IP cameras, each one collecting 2 Mpixels frames at a rate higher than 20 frames/sec. After an approximate temporal synchronization of the video streams and the definition of an homography to link the ground points in every views, the video data have been augmented through the definition and collection of metadata. Those metadata are expected to support efficient browsing of the content, and automatic personalized summarization of the event captured by the distributed set of cameras. Metadata include (1) low-level feature distributions, (2) object trajectories, and (3) events of interest attributes. The data and metadata are publicly available upon email request to the APIDIS partners.

Keywords: Multi-camera network, metadata, video analysis.

1 INTRODUCTION

In today's society, content production and content consumption are confronted with a fundamental mutation. Two complementary trends are observed. On the one hand, individuals and organizations become more and more heterogeneous in the way they access the content. They want to access dedicated content through a personalized service, able to provide what they are interested in, when they want it and through the communication channel of their choice. On the other hand, individuals get easier access to the technical support and facilities required to be involved in the content creation and diffusion process. The success of video sharing websites like YouTube illustrates the emergence of such an individual and personalized implication of end-users in the task of producing and making video content available.

In this paper, we describe the approach of the APIDIS consortium to capture and annotate video content, in a way that participate to the future evolutions of the content production industry towards automated infrastructures allowing content to be **produced,**

stored, and accessed at low cost and in a personalized and dedicated manner.

Figure 1 depicts the APIDIS vision. Content is captured and produced automatically, without the need for costly handmade processes. In a typical application scenario, the APIDIS acquisition sensor network is composed of microphones and cameras, which in the example cover a basket-ball court. Distributed analysis and interpretation of the scene is exploited to decide **what to show about an event, and how to show it**, so as to produce a video composed of a valuable subset from the streams provided by each individual camera. The system provides a solution to cover local (sport) events at low cost as no technical team or cameraman is needed. More generally, the system can be used to report events that involve human activities, for example in surveillance scenarios.

To achieve cost-effectiveness, APIDIS relies on:

- Automation of the production, based on scene analysis, to prevent or reduce human intervention in the content creation process;
- Exploitation of omnivision and distributed sensing to cover large areas with a limited number of static sensors. The static nature of sensors adds to cost-effectiveness because it permits to store all relevant content and to process it off-line. In contrast, the use of moving PTZ cameras, automatically controlled to focus on the actions-of-interest in the scene, would require real-time processing and interpretation of the captured data.

As a consequence of its cost-effectiveness, APIDIS aims at keeping the production of content profitable even for small- or medium-size targeted audiences. Thereby, APIDIS promotes the emergence of novel markets, offering a large choice of contents that are of interest for a relatively small number of users (e.g. the summary of a regional sport event, a university lecture, or a day at the nursery).

In addition, by automating the production, APIDIS promotes the collection and supply of intelligent digital content. Intelligence refers here to the identification of salient segments within the content, and to the exploitation of that knowledge to adapt and personalize content summarization according to the individual user needs. In that sense, the APIDIS autonomous production also enables content access personalization.

Generating a personalized summary simply consists in (re-)running the production process with input parameters corresponding to the specific constraints expressed by the user.

From the above description, we conclude that content acquisition has to meet a double objective. First, it should capture the information required to analyze and interpret the scene at hand. Second, it should support nice-looking and informative rendering. In this paper, we describe a practical setting to meet both requirements.

A distributed architecture involving high-resolution conventional and omnidirectional sensors is presented in Section 2, which was deployed to capture content in the context of a basket-ball game. Both raw content and calibration parameters are made publicly available upon email request. In Section 3, we describe how the content is augmented through semantically relevant metadata, including identification of events-of-interest, object trajectories, and foreground object localization. This material provides a useful content and ground truth data to demonstrate and benchmark distributed-video analysis algorithms.

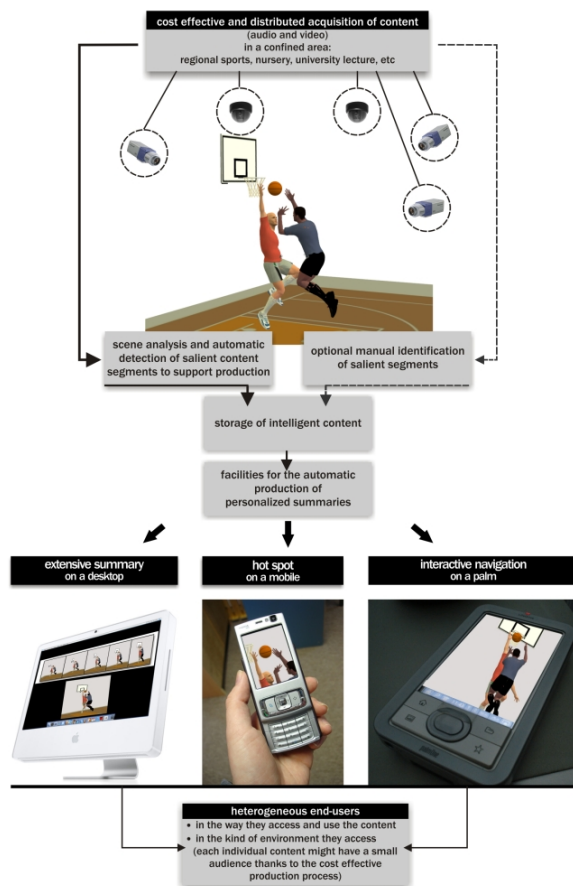


Figure 1: The APIDIS vision. Identification of salient segments within the content captured by a camera network to support personalized and autonomous content production.

2 ACQUISITION SETTING

This section describes the system that has been considered to capture a set of complementary images around a basket-ball field. The objective was to capture images that could support both efficient analysis and nice-looking rendering of the event. Only stationary video cameras have been considered. They were connected to a central server.

2.1 Camera and server specifications

The APIDIS acquisition system is composed of several high-resolution cameras distributed around the sport field. The main criteria for the selection of the cameras were the resolution, the frame rate, the sensibility and the cost of the overall system. The main features of the selected Arecont Vision AV2100 cameras depicted in Figure 2 are: 2 Mpixel IP-cameras sensitive to 0.1 lux at F1.4 providing 24 fps at 1600x1200 and featured with a captor size of $\frac{1}{2}$ inch.

For two cameras providing a top view of the basket-ball field, Fujinon wide angle lenses have been used, see Figure 3. When installed with the AV2100 cameras, they provide horizontal and vertical view angles of $136^{\circ}18' \times 102^{\circ}19'$ for a focal length of 2.7 mm. The other cameras use more standard lenses.

The server that collects all the video streams is a Hewlett Packard DL380 G5 with Intel(R) Xeon(R) CPU E5420 at 2.5 GHz with 2 GB of memory and two 73 GB disks. In the course of the APIDIS project, this machine will also be used to serve images and metadata to the algorithms in charge of the automatic generation of sport events summaries.



Figure 2: Arecont Vision AV2100 color, 2 Mpixels.



Figure 3: Fujinon FE185C086HA-1 super wide angle lens.

2.2 Camera positioning

To preserve consistency between the movements rendered in all views, cameras have all been distributed on the same side of the axis joining the two baskets. In order to limit the number of cameras to deploy, some asymmetry has been introduced and more cameras have been placed on the left-hand side of the field. Figure 1 displays all seven camera views at the same time. The first two super-wide-angle pictures provide top views, one for each side of the basket-ball court, with an incidence angle of about 45°. The last three pictures provide zoom in views on the left-hand side of the field¹. Those three zoom in views are particularly interesting for rendering, while large angle views allows for consistent tracking along the game.



Figure 1: Snapshots from the 7 synchronized cameras



Figure 2: Top view with a super-wide-angle lens

¹ In a complete system, it is foreseen that the same configuration of three cameras is installed on the other side of the court.

2.3 Storage format

The seven video streams are recorded on the server in their native MJPEG format. The bandwidth and required disk space is about 300 MB per camera per minute. The files are organized with a directory for each camera and a file for each minute of video sequence. A file containing the timestamp of each frame (see Section 3.1) is also recorded with each video file.

The metadata associated to specific cameras (e.g. foreground mask, defined in Section 3.5) are also distributed in the respective directories, while metadata relative to the game (e.g. clock and non-clock events, defined in Section 3.3) are stored in the upper directory. All filenames describe the source, the timestamp of the first element they contain and a media type extension. An example is *cam1.20080405T131900Z.mjpeg*.

This directory tree structure and filenames convention allow fast search of files for browsing.

3 METADATA DEFINITION AND GENERATION

This section defines the set of complementary metadata that have been considered to identify salient segments within captured video content, thereby supporting efficient rendering and summarization of the basket-ball game.

3.1 Input data: synchronized multi-views video streams

The main assumption underlying the annotation format is the existence of a common unique temporal and spatial reference for all camera views, so that all information defined relatively to one camera can be mapped to the absolute spatial and temporal coordinates of the scene at hand.

A **common and unique time reference for all camera views**, could obviously be obtained by synchronizing the instants at which frames are captured by the cameras. However, such synchronization capability requires expensive professional firewire or GigaEthernet cameras. In contrast, each IP camera from the setting described in Section 2 captures as much frames as possible, and sends them to a common server. When it receives a frame, the server stores it, and labels it with a timestamp, corresponding to the instant of arrival. Hence the timestamp refers to the server clock, which is common to all cameras, but corresponds to the storage instant rather than to the instant of capture. In this context, we propose to build a common reference time line for all camera views as follows. We sample the server clock at 20 Hz, and for each time sample and each camera view, we select the frame with the larger timestamp below the instant of interest. Doing so, we generate a regular stream of multiple frames, each frame being a reasonable

approximation of the signal that would be captured by the corresponding camera view at the instant of interest. This multi-view stream is the one considered for subsequent processing and annotation.

Regarding the spatial correspondence between views, we rely on calibration parameters and define an **homography** to link the points of the basket-ball court in each camera view, thereby defining a **single referential** for all objects in the scene.

Note that the annotation does not assume overlapping views, nor a complete coverage of the captured scene.

3.2 Metadata : event, trajectories, and low-level features distribution

The annotation process consists in the collection of three kinds of complementary metadata:

1. Events of interest that identify specific actions of the game, and are characterized by a number of attributes (type of event, time instant, position, player involved).
2. Object trajectories that define the position of a given object (a player or the ball) within a finite period of time.
3. Efficient representations of the spatial (and potentially temporal) distribution of low-level features, typically the foreground mask or motion vectors fields, which support scene rendering.

Each type of metadata is further defined in the rest of the section, together with a description of the mechanisms to collect them.

3.3 Clock- and Non-clock- events

In the following, an event refers to a game action or a player/spectator(s) behaviour that can be inferred based

on the analysis of the signal captured by the sensor network, mainly composed of cameras at this stage of the project.

The events potentially relevant to identify salient segments of a basket-ball game have been listed and defined through the XML hierarchical syntax depicted in Figure 6.

Two categories of events are of interest:

1. Events that have a direct impact on the 24'' clock of the basket-ball game. Therefore, we name them the *clock-events*. They correspond to the events associated to the starting, stopping or re-initialization of the 24'' clock (i.e., when the game is interrupted or when the ball hits the basket or is gained by the opponent team).
2. Events that do not cause any specific action on the 24'' clock. They typically are displacements and interactions of players during the game, or some subjective interest expressed by spectators/viewers about the game.

We decided to differentiate the clock and non-clock events because the 24'' clock is easy to monitor in an automatic way and is closely linked to the semantic of the game, or at least to all objective statistics that are generally collected by coaches and players about the game (scored points, fault, and so on). Hence, for clock-events, the objective of the annotation tools is reduced to recognizing and characterizing the event associated to the change of clock state.

In contrast, non-clock events refer to global or individual behaviour in the game, and do not have specific and objective time anchors. They are thus more challenging to detect, and are mainly considered to support the manual introduction of complementary information about the game. The APIDIS objective is not to detect and characterize non-clock events in an exhaustive and automatic way, and only partial hints about some of those events is provided.

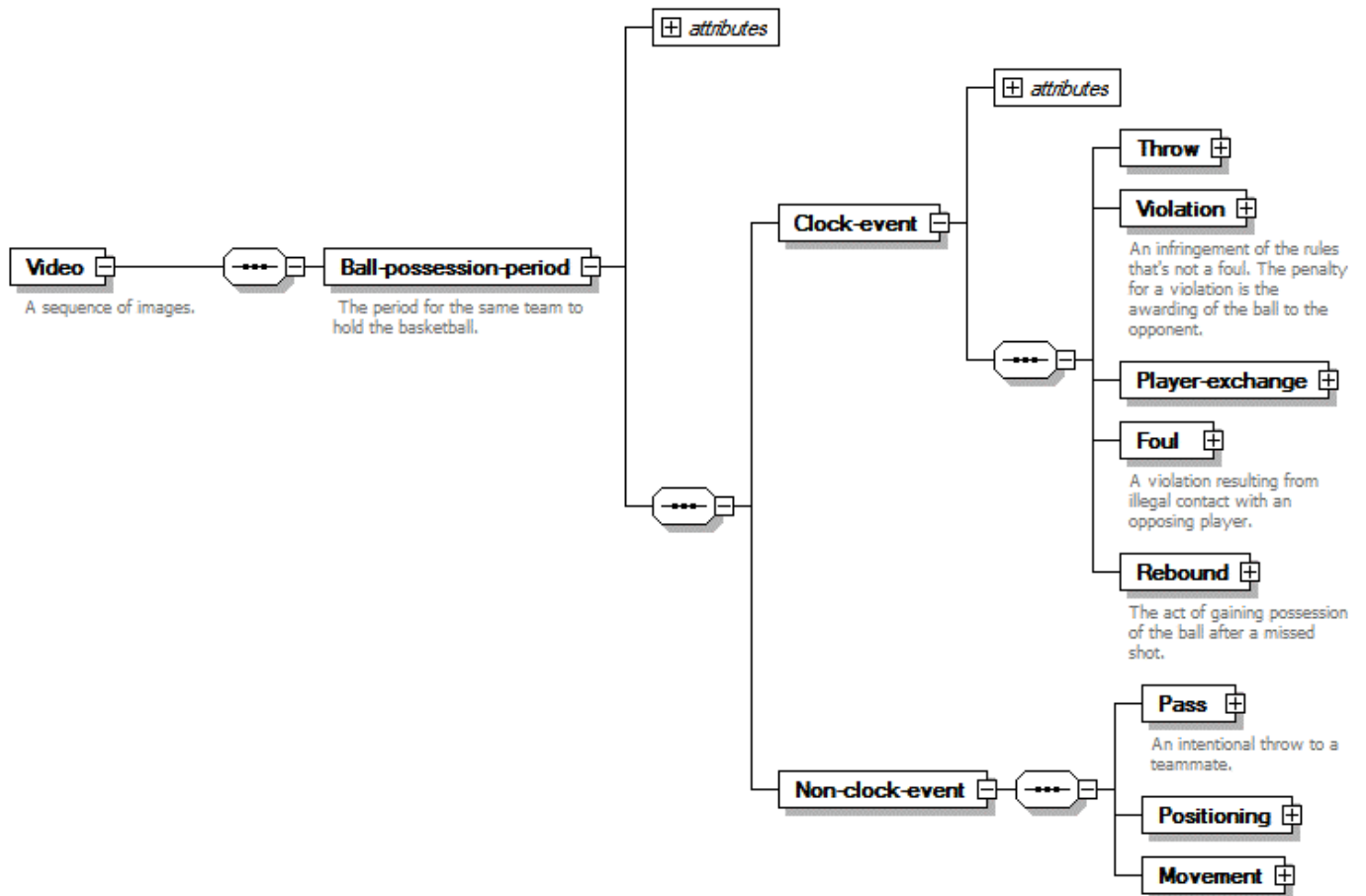


Figure 6: XML file format for basket-ball event annotation.

We observe in Figure 6 that events are grouped in subsets corresponding to ball possession periods, which are expected to be meaningful in a summarization context. In practice, since the clock is initialized each time a team gains the ball, the period of time between two clock events is characterized by a single attacking and defending team. A ball possession period can thus easily be defined by merging adjacent periods between clock-events for which the same team is attacking/defending.

In more details, the attributes of the ball possession period typically include start- and end-times, team label, and optional information about the behaviour of offending and/or defending teams (fast-attack, zone defence, press, etc.).

In contrast, the attributes of a clock or non-clock event include its timestamp in the camera time and game time² references, plus a set of attributes that are specific to the event type (e.g. throw, fault, etc). As an example, Figure 2 presents the list of attributes associated to a fault event.

An exhaustive description of the attributes associated to each type of event is provided on the APIDIS web site (<http://www.apidis.org>). Typically, the attributes identify the players involved in the event, define a time frame for the event, and refines the nature of the action at hand through a number of options (e.g. foul-type, throw-type).

At the current state of the project, an interface has been developed to support manual definition of clock- and non-clock-events. The annotation XML file is publicly available, together with the video streams and calibration parameters. Within APIDIS, it is used to initiate the automatic personalized summarization mechanisms, and serves as a ground truth reference to validate the video analysis/recognition tools that are developed to generate those metadata automatically.

² Here, the game time refers to the clock aggregating the actual time played in the game. The game time can be computed based on the timestamp associated to clock events, since the game clock is stopped/started each time the 24'' clock is stopped/started.

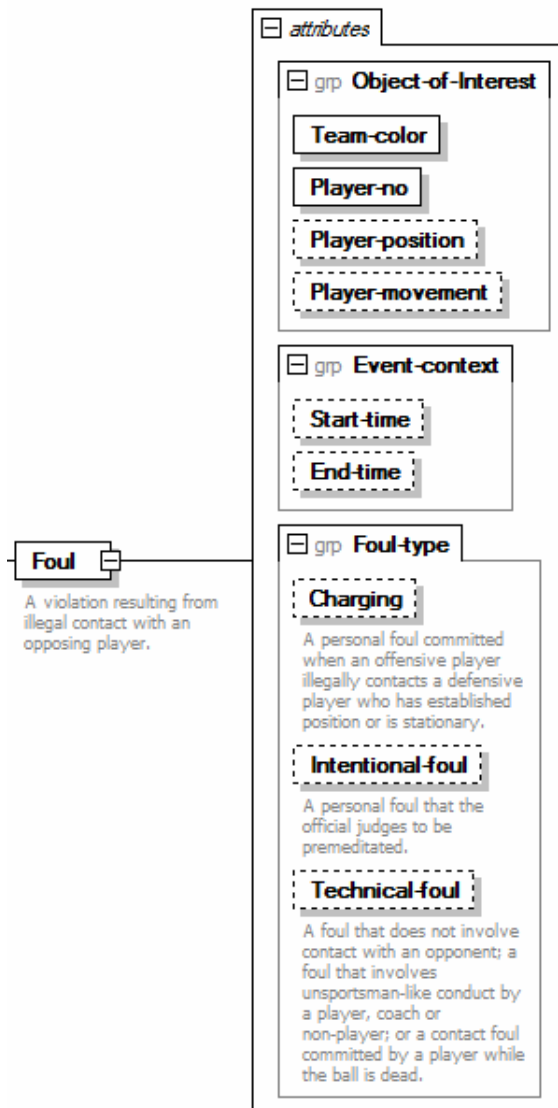


Figure 7: List of attributes associated to a fault event.

3.4 Object trajectories

In our sport-event analysis framework, object trajectories define the position of a given object (a player or the ball) within a period of time. It is worth mentioning that the trajectory does not necessarily track the object in a continuous manner all along the game. Instead, the file collecting the trajectories of an object consists in a set of potentially interrupted tracks. Hence, those metadata files can be progressively incremented, based on automatic detection and tracking of objects within the scene.

In a first stage, an object is detected and tracked without being identified. In a second stage, a recognition engine parses the object bounding boxes, so as to recognize it automatically, e.g.

by recognizing the number of the player from the image captured at a well-chosen instant and from a well-chosen viewpoint along the trajectory. Joint processing of multiple views is exploited to detect objects and define their trajectories.

The first release of object trajectories metadata is based on conventional particle filters, and on manual identification of the player associated to most relevant trajectories. Advanced tracking methods and automatic player recognition modules will be developed in the course of the project, and should result in updated and completed release of those trajectory files.

3.5 Foreground mask and motion vector distribution

Here, we consider the characterization of low-level features within each individual camera views. The envisioned features define the location of foreground objects within the scene, and could potentially be augmented based on motion vector fields computation.

In APIDIS, those features are considered to fill in the lack of information resulting from the potential interruptions of object trajectories, when selecting rendering parameters. In other terms, those features are expected to support the selection of the appropriate camera view and cropping parameters when rendering a given period of the game.

At the current stage of the project, only the extraction of foreground objects has been considered. A sub-sampled mask of foreground areas has been defined automatically for each frame of each camera view, based on the subtraction of a background Gaussian mixture model. The process is similar to the one described in [2-3].

4 CONCLUSION

This document has presented the acquisition setting and annotation methodology considered by the FP7 APIDIS project. As practical outcome it provides public access to multiple high-resolution video streams captured by a network of 7 cameras distributed around a basket-ball field, and to the set of metadata ranging from low-level scene descriptors to ground-truth high-level semantic concepts.

5 REFERENCES

- [1] <http://www.apidis.org>
- [2] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 246–252, June 1999.
- [3] X. Desurmont, C. Chaudy, A. Bastide, C. Parisot, J.F. Delaigle and B. Macq. Image analysis architectures and techniques for intelligent systems. In *IEE proc. on Vision, Image and Signal Processing, Special issue on Intelligent Distributed Surveillance Systems*, 2005.