Official URL

DOI : https://doi.org/10.1109/RCIS.2018.8406677

Modeling Metadata of CCTV Systems and Indoor Location Sensors for Automatic Filtering of Relevant Video Content

Franck Jeveme Panta
IRIT
Université Paul Sabatier
Toulouse, France
franck.panta@irit.fr

Geoffrey Roman-Jimenez
IRIT
Université Paul Sabatier
Toulouse, France
geoffrey.roman-jimenez@irit.fr

Florence Sedes
IRIT
Université Paul Sabatier
Toulouse, France
florence.sedes@irit.fr

***Abstract*—Location sensors and Closed-circuit Television (CCTV) cameras are widely used for the surveillance of people, objects and areas. These devices (sensors, CCTV cameras, etc.) generate a large amount of heterogeneous data, making their analysis and management difficult and very time-consuming. In the context of video-surveillance, automatic extraction of the relevant information among the mass of multi-sources information produced by these systems could significantly reduce the investigation time and facilitate their analysis. In this paper, we propose an approach that combines data from Indoor Location Sensors and metadata from CCTV cameras to automatically retrieve relevant video segments during research of evidence accident or crime events in indoor environments. The proposed method consists in i) the reconstruction of the mobile device trajectories from indoor location sensors and ii) the identification of the CCTV cameras intersecting the reconstructed trajectories. To ensure industrial transferability, indoor location sensors data were embedded in a generic model of CCTV cameras metadata that instantiates the standard ISO 22311 (relative to digital video-surveillance contents). We provide an experimental evaluation demonstrating the utility of our approach in a real-world case. Results show that our method helps the CCTV operators to effectively retrieve the relevant video and drastically reduce the time of analysis.**

***Keywords*—CCTV System, metadata, spatio-temporal, trajectories, tracking.**

I. INTRODUCTION

The search for *a posteriori* evidence is a very active research topic that has been ”expanded” in Closed-circuit television (CCTV) Systems. As detecting crime in real-time is not always possible, CCTV systems are often used as resources for the gathering of robust-evidence during investigation [1]. Nevertheless, video analysis remains challenging due to the large volume of videos generated by CCTV cameras requiring an expensive and time-consuming process to analyze its content. During the last decades, major advances have been made in pattern recognition to analyze the video content and automatically perform task such as object detection, object tracking, person identification, event detection, etc. [17]–[20]. However, the intense computational nature of these algorithms, often developed on specified conditions, can limits their use for real-world cases [17], [19]. In this context, one way to overcome these limitations can be to avoid processing unnecessary data by filtering them using information provided by their metadata, and thereby reducing the processing-time by focusing only on relevant content.

In this paper, we address the problem of analyzing large amount of data generated by CCTV systems in the specific context indoor environments. The goal is to significantly reduce video ”manual” analyze time by providing to CCTV Operators a list of cameras that may have captured the trajectories of potential suspects with the corresponding video segments (time interval). To do so, we propose to aggregate multi-source metadata, which provide complementary information, to facilitate the management and the retrieval of relevant video. The proposed method consists, first, in the trajectories reconstruction of mobile objects in an indoor environment using the positions generated by Indoor Location Sensors (ILS). In this paper, Ultra-Wide Band (UWB) sensors were used to generate the mobile objects positions. Then, the reconstructed trajectories help to query both technical and material metadata of the CCTV system. These metadata describe the movement and field of view of the cameras and are used to automatically retrieve the relevant video segments (i.e. the video segments intersecting the trajectory of the sought object). To ensure industrial transferability of our work, we modeled the metadata provided by CCTV cameras as an instantiation of the standard ISO 22311 [16] on which we embedded a modeling of the metadata provided by ILS.

This paper is organized as follows: Section II presents a brief summary of related work; Section III describes the method that we propose to automatically retrieve relevant video segments; Section IV presents an implementation of the proposed method; Section V shows an real-case experimentation that we performed to evaluate the capability of our method to retrieve relevant video content. Finally, in section VI, we discuss and conclude with suggestions for future research.

II. RELATED WORK

A. Spatio-temporal filtering using trajectories

In the literature, some approaches address post-processing of data that highlight relevant elements (persons, objects, etc.). Video shots search by reconstitution of trajectories is addressed in [2], [3]. Authors have proposed a method which, starting from trajectory segments, selects the cameras that may have filmed the given segments and identifies the video sequences corresponding to each camera according to the inferred locations and time interval. The context is an outdoor environment with public transportation and road networks. In indoor environment (context of this paper), the positioning system is different and more complex to model. The querying of spatio-temporal data from trajectories is discussed in [4]. Authors suggest to investigators, the cameras that may have filmed the suspect's trajectories. These trajectories are constructed based on information gathered from witnesses or victims during investigations. In our approach, we propose to automatically build the trajectories of potential suspects using location sensors installed in the buildings.

B. ISO 22311: International Standard for Societal security and Video-surveillance

The International standard ISO 22311:2012 [16] specifies standard and describes good practices to ensure interoperability for the collection and the use of information and data extracted from video-surveillance systems. The standard specifies non-exhaustive structuration of the metadata provided by CCTV cameras (Sensor Description, Time information, Optical information, Device Location, etc.).

In this work, we implemented a description of sensors metadata described in the standard ISO 22311:2012 and accordingly modeled the metadata collected from CCTV cameras to facilitate query processes and information retrieval, and also ensuring the industrial transfer of our work.

C. Architecture of an Indoor Positioning System

An Indoor Positioning System (IPS) is a system that locates objects or people inside a building using radio signal, magnetic fields, acoustic signals or other sensory information collected from mobile devices. According to [11], IPS generally consists of three subsystems: (1) the Sensor subsystem, (2) the Interface subsystem and (3) the Database subsystem (Figure 1). In a normal situation, transmitters continuously broadcast their signal in the coverage. Any device equipped with special sensors in their coverage will receive a signal. The received signals will then be processed by a central processing unit (where the positioning algorithm is installed) before being compared with data collected from the database server. Finally, the exit of the system will display the different positions of objects on the building plan for example.

In this work, we use the metadata generated by IPS to reconstruct the trajectories of mobile objects within an indoor environment, and combined them with the from CCTV metadata.

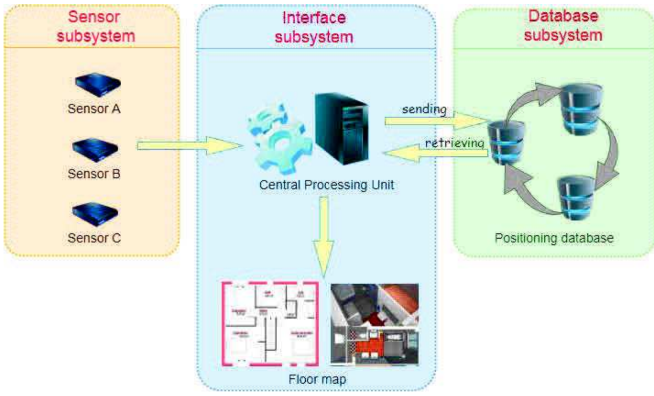

Fig. 1. Indoor Positioning System architecture. This architecture describes the interaction between the different components of an IPS

D. Types of spatial queries appropriated to the indoor environment

A spatial query is any user's query that includes at least one geolocation or spatial location element (expressed in terms of geometric coordinates or symbolic names) and/or a time reference (date, time, and interval) [5]. The different types of queries are described below:

- "Position queries": position queries return locations of static and moving objects, and are processed according to a geometric or symbolic model. Example: "Where is the server room?". This type of query is the basis for any other spatial query that uses location information [6].
- "Range queries": range or region queries are used to search and retrieve information about an object in an area specified by the user. Specifically, this type of query returns information about objects located in a given user-defined area or perimeter (usually this area is defined w.r.t. a reference object, e.g. the circle of radius 1km having as its center the position of the object): "what are the rooms on floor 8?", "what are the printers located within 30 meters of my position?". The regions may be characterized by a circular or rectangular window in which the objects of interest must be located [8]. In addition, region queries can be static or dynamic. Similarly, a region query can be applied to static or dynamic data, depending on whether the target objects are moving or not. Region queries are used in Geographic Information Systems such as Google Maps, Bing Maps and mobile navigation [7]. A large number of studies have been performed to handle this type of query effectively and accurately [14], [15].
- "Navigation queries": trajectory or navigation queries consider object tracking but do not consider the position or region around an object at a given time [9]. They directly help users to find points of interest by providing trajectory information while optimizing certain criteria such as total distance or time to travel. Example: "What's the shortest path to room 2?"

Fig. 2. Example of tracking devices.

- “k Nearest Neighbor queries (kNN)”: kNN queries return the closest k objects to a given object. In opposition to region queries, kNN queries are independent of distance. The user initiates a query by specifying some characteristics on objects of interest so that the closest objects whose specifications match these characteristics are retrieved [10].

In the proposed approach, we use ”Position queries” for the devices tracking and trajectories reconstruction and ”Range queries” to retrieve all devices located in a given area.

III. PROPOSED APPROACH

This section develops our approach for automatic retrieval of video from a CCTV system during research of evidence in accident or crime event. The main goal is to reduce the amount of data that requires to be analyzed by the CCTV operator in order to reduce the time of analysis and help the investigators. The proposed method can be summarized in the following steps: (1) modeling tracking data generated by ILS; (2) reconstruction of object trajectories; (3) querying process (“Intersection” of reconstructed trajectories with camera metadata in order to find video segments that may be relevant for investigations).

A. Tracking data modeling

We considered Ultra-wide band (UWB) for trajectory reconstruction. UWB sensors are accurate for indoor positioning, robust to impact of building infrastructures and to errors caused by interference and signal reflection [1]. Figure 2 shows an example of tracking devices mounted on a building map. The dots represent the different positions of each device (related to objects/persons) detected by the UWB sensor.

The data related to UWB detection are described as follows:

- $DeviceID$: identifier of each detected device (MAC address).
- $Feature$: characteristics of each object.
- $DateTime$: date and time of detection.

Event : **Museum robbery**

Location/Address : **“XX Boulevard XXX, 7XX3 Paris”**

Additional address : **Floor 2 / Room 204**

Date and time : **17/10/2017 between 4pm and 5:42pm**

Distinctive marks / Clothes color / Robot portrait

Fig. 3. Example of query.

- $Position(X, Y, Z)$: position of the detected object. X and Y are the coordinates in the building plan and Z is the floor number.

These location information have been collected and compiled in XML files. We implemented a parser that allows to read the structures of these files, and performed queries that enable to store the contents in Oracle database. For handling geometric objects, we use Oracle Spatial extension which has more advantages than the other spatial databases, namely:

- Integration of native support for all geospatial types and models such as vector and raster data, topological and network models (to address the needs of highly specialized GIS).
- Stability and performance.

Note that we consider here that the location information are not noisy (this paper does not discuss problems related to position detection, but uses the positions generated by an appropriate system to rebuild trajectories).

B. Trajectories reconstruction

Relevant information for an investigation may include: the location, date and time interval of the incident, information that allow identification of the suspect (if available). Figure 3 shows an example of a query formulation.

This query is formalized in a JSON file for easy interpretation. The defined query model has two main parts: a spatial part and a temporal one (Figure 4).

The location information and time interval given in the query allow to build the trajectories of the localized devices. To do this, we have implemented an algorithm that runs in two steps:

1) First step: Objects selection: This step consists of selecting all stored and detected devices of which we want to build the trajectories. To do this, the identifiers of all devices in the area (surface) concerned by the query are selected. The result at this step is a list of devices (each device identified by its $DeviceID$).

2) Second step: Objects tracking: For each device in the list, all the stored positions during the time interval of the query are searched. The trajectories of each device are accessed by constructing a poly-line that chronologically links all the positions found.

Algorithm 1 presents the trajectory recovery process. The inputs are: (1) the area (surface) where the incident occurred,

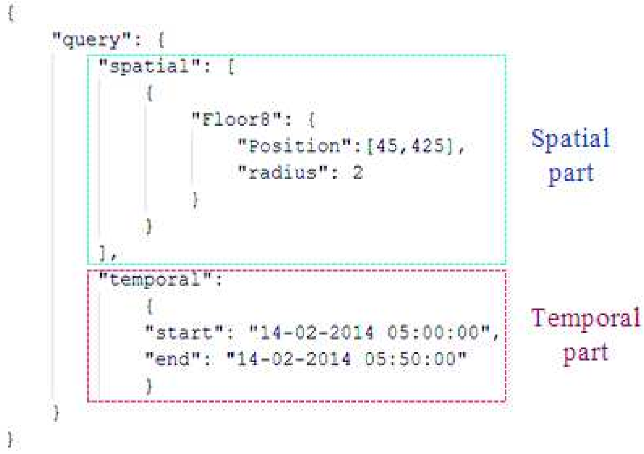

Fig. 4. Example of a JSON query.

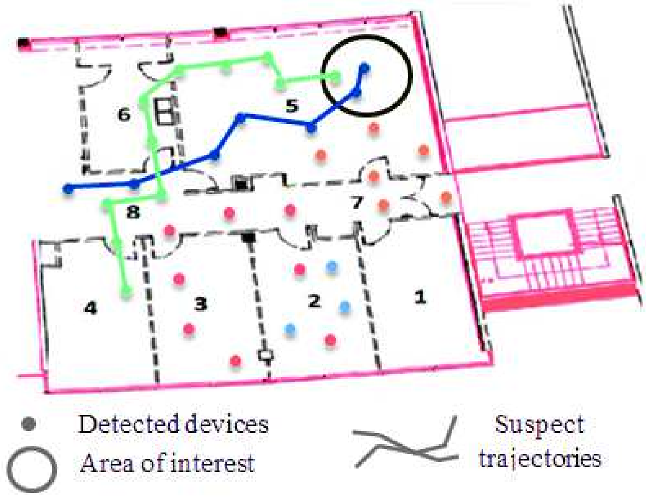

Fig. 5. Examples of reconstructed trajectories.

which is represented by the center circle poi and the radius ray, and (2) the time interval of the query $[t_{start}, t_{end}]$. The function $select\_deviceID$ takes an area as a parameter, and returns the list of devices in that area. The $select\_position$ function takes a device and a time interval as parameters, and then returns the list of positions of that device during that time interval. The function $build\_polyline$ builds a poly-line with a list of positions set as parameter.

Algorithm 1: Trajectory reconstruction procedure

Input: $poi,\ ray,\ [t_{start}, t_{end}]$
Output: A list of trajectories and their segments
$roi \leftarrow build\_geometry(poi, ray)$;
$listOfDevices \leftarrow select\_deviceID(roi)$;
foreach $device_i$ *in* $listOfDevice$ **do**
  if $DATE\_TIME(device_i) \geq t_{start}$ *and* $DATE\_TIME(device_i) \leq t_{end}$ **then**
    $listOfFilterDevice \leftarrow device_i$;
  end if
end foreach
foreach $device_i$ *in* $listOfFilterDevice$ **do**
  $listOfDevicePosition_i \leftarrow select\_position(device_i, [t_{start}, t_{end}])$;
  $polyligne_i \leftarrow build\_polyligne(listOfPosition_i)$;
end foreach

Examples of reconstituted trajectories for two detected suspects in a given area are shown in Figure 5.

C. Querying process

In this section, we describe the metadata (related to the camera) used to query the constructed trajectories and the querying process.

1) Metadata description: The efficiency of any CCTV system depends on the performance of the cameras and their installation (position, settings, etc.). With technological development, the installation of cameras can be efficiently implemented in terms of fixed platforms or any mobile device. In this paper, we focus on fixed cameras installed in indoor environments (metro stations, supermarkets, shopping malls, airports, etc.). The captured scenes change with camera rotation.

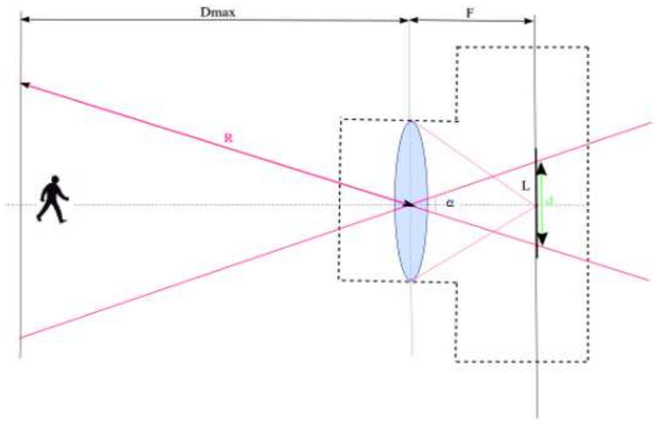

Fig. 6. Camera field of view.

- *Camera field of view*: A camera located at a given position, with a specific orientation and installation can capture a given area. The field of view represents the area of the scene seen by a camera. The field of view is vertical and horizontal. This area is impacted by a number of parameters related to each camera. Figure 6 illustrates the field of view and its parameters (shown in Table I).

Camera installation: For a greater field of view accuracy, the camera installation can be considered. The height and tilt angle of the camera determine the maximum length that can be seen by the camera (D_{max}) and the shadow area (D_{min}) that cannot be seen by the camera. The installation can be changed over time so we will define the installation of a camera as a triplet (inclination, height, time) vector. Figure 7 shows the installation of a camera with the parameters (TableII) that impact the area observed by the camera.

The metadata for the field of view and camera location are described respectively in Table II and Table III.

TABLE I

PARAMETER OF THE CAMERA FIELD OF VIEW.

Parameters	Description
α	Angle of view
d	Sensor size
F	Focal distance of lens
D_{max}	Maximum visibility length
D_{min}	Minimum visibility length
R	Side of the triangle formed by the field of view
L	Classic picture size
H	Camera installation height
Θ¸	Camera tilt angle
H_{min}	Height of the target object

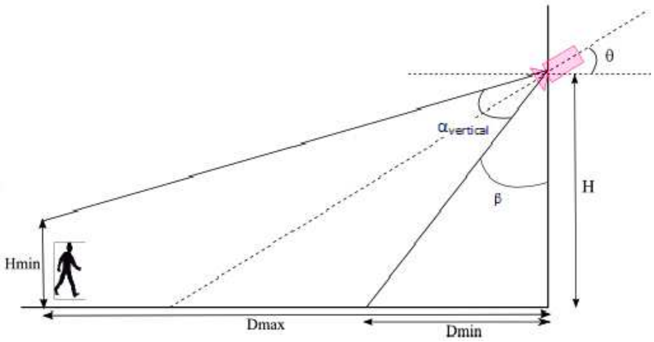

Fig. 7. Camera installation and parameters.

2) Data modeling: Here, we present a generic data model that provides a structured combination of the collected data that we used to facilitate the query processes and information retrieval. This data model is an instantiation of the standard ISO 22311:2012 [16] on which we embedded the data from ILS, ensuring that industrial integration of our work will provide cross-system compatibility.

Figure 8 shows a class diagram of the proposed data model. The framed part of the model corresponds to the modelization of the data generated by ILS.

Because it is generic, the proposed model is extensible for integration of other information and can be used in further real-cases, or other problems related to CCTV.

TABLE II

METADATA DESCRIBING THE FIELD OF VIEW OF CAMERAS.

Label	Comment	Filling method
Sensor size	value (double)	Manual / Automatic
Focal length	value (double)	Manual / Automatic
Angle of view	Value (degree)	Automatic (computed)
Pan, Tilt, Zoom, Roll	value (double) or value range	Automatic (computed)
Installation height	value (double)	Manual

TABLE III

CAMERA LOCATION METADATA.

Label	Comment	Filling method
Reference system	String (characters)	Manual / Automatic
Geometric position (2D / 3D)	2D / 3D coordinates	Manual / Automatic
Symbolic position	String (characters) or numeric value	Manual / Automatic

3) Querying: As presented in Section 3.2, relevant data for the investigation are the spatial information and time interval of the query. This information allowed us to detect interesting potential devices and rebuild their trajectory.

4) Description of the hasSeen operator: The goal is to offer for CCTV operators, video sequences that may contain interesting images for the investigation (suspects, trajectories, etc.). To do this, it is necessary to search for cameras whose field of view (which can be variable) intersected the trajectories built in the given interval $[t_1, t_2]$. Using the $hasSeen$ operator defined in [2], [12] as follows: given a spatial trajectory composed of segments $t_r = (u_1, \ldots, u_n)$ and the time interval $[t_1, t_2]$. $hasSeen(t_r, t_1, t_2)$ returns the set of cameras $C_i (1 \leq i \leq m)$ that captured a segment $u_k (1 \leq k \leq n)$ and a video sequence between two moments t^i_{start} and t^i_{end} within the interval $[t_1, t_2]$. We do not consider the different segments that constitute trajectories, because in an indoor environment, the segmentation of the trajectory is not relevant. The operator takes the full trajectory as a parameter. The result is all video segments of the cameras that may have filmed the desired scene.

$$
\begin{aligned}
hasSeen\ &: u_1, u_2, \ldots, u_n, [t_1, t_2] \implies Z \\
Z = \{ &C_1 :\ t^1_{start} \rightarrow t^1_{end},\ u_k (1 \leq k \leq n), \\
&C_2 : t^2_{start} \rightarrow t^2_{end},\ u_k (1 \leq k \leq n), \\
&\ldots \\
&C_m : t^k_{start} \rightarrow t^m_{end}, u_k (1 \leq k \leq n) \}
\end{aligned}
$$

5) Camera selection algorithm: Algorithm 2 presents the camera selection process. All cameras are fixed, but have fields of view that can change. The result of the algorithm is the set of pairs $R = (C_i, [t_a, t_b])$, C_i is one of the installed cameras, t_a, t_b are timetamps such as $t_1 \leq t_a \leq t_b \leq t_2$. The defined operator checks which are the cameras whose fields of view intersected the built trajectories and between which time intervals (t_a and t_b). The algorithm proceeds in two steps: the candidate selection step (purely spatial filtering) and the results refining step (temporal aspect):

- The filtering step applies an algorithm corresponding to a "Region Query" type similar to the one presented in [13]. It allows to select for each trajectory, the cameras located at a distance less than or equal to the maximum visibility distance of all existing cameras in the database. This avoids the evaluation of the spatial intersection (expensive operation) for fields of view of the cameras which are located at a distance that makes it impossible to see the queries trajectories. The filtering is executed by the function: $camDisExtract(t_{r_n}$ $, max(visibleDistance(fov_j)))$, where t_{n_r} represents the reconstructed trajectories and $visibleDistance(fov_j)$ gives the maximum visibility distance of a field of view of a camera.

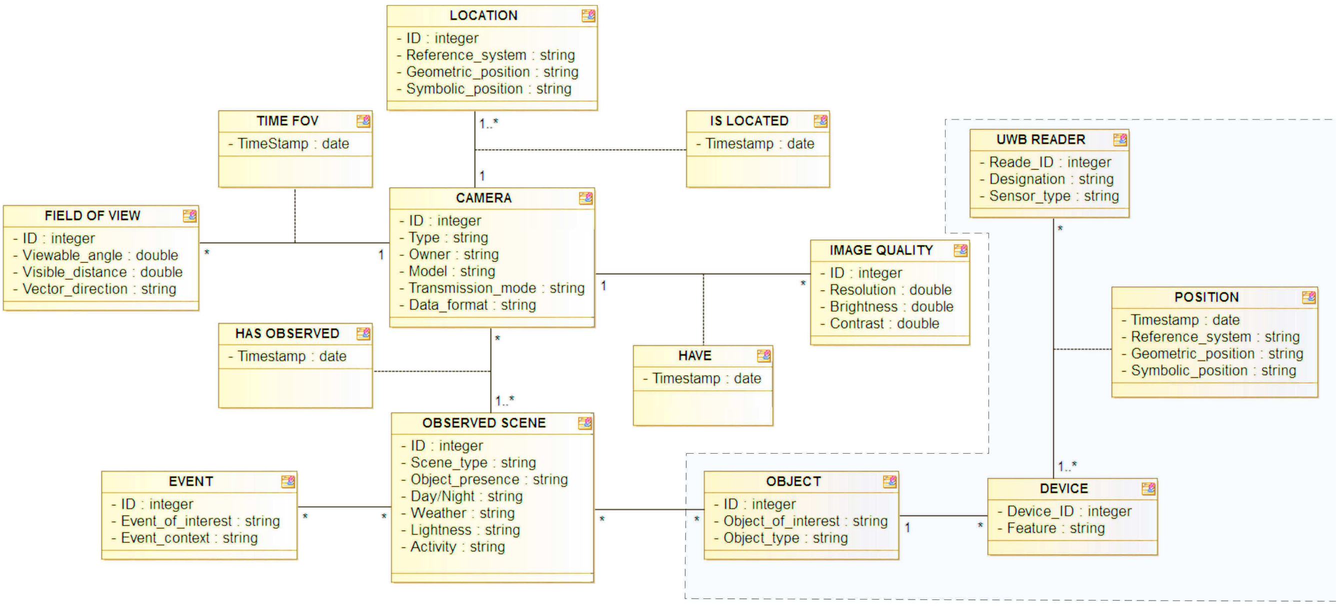

Fig. 8. Model used for the metadata collected from CCTV cameras and indoor location sensors (ILS). The model is an instantiation of the standard ISO 22311:2012 embedding additional metadata collected from ILS (framed part).

- For the cameras that have been selected for filtering, the refining step calculates the geometry of the field of view ($build\_polygon(fov_j(c_j))$) concerned by the time interval of the query.

IV. Prototype Architecture and Implementation

A. Architecture

The proposed prototype is based on a multi-layered architecture. Specifically, we use three-level architecture (see Figure 9) with different modules.

1) The "View" Layer: Consisting of the main components of the user interface, it groups together the bricks that perform user interaction and visualization of results and data. It allows the user to enter queries, view data and results.

2) The "Logic" layer: It contains the main modules of the system such as (1) the trajectory builder which allows to follow the different paths of the potential suspects, and (2) the Search Engine which implements the proposed algorithms. The **Trajectories builder** module interprets queries entered by users and uses location information saved in the Oracle database to rebuild the trajectories of selected devices as presented in algorithm 1 (Path retrieval). Algorithms that allow metadata to be queried (technical, motion, camera field of view) from built trajectories are implemented in the **Search Engine** module. The reconstructed trajectories are stored in the database; the advantage here is to index these trajectories later using spatial indexes [8]. A spatial index allows to speed up the access to data (e. g. find objects having a spatial relation with a point or a research area, find a pair of objects with spatial interaction).

Algorithm 2: Selection of cameras that intersected the query trajectories

Input: Trajectories and the time interval $[t_1, t_2]$
Output: List of cameras that have filmed the trajectories and corresponding time intervals

foreach tr_n *of the query* **do**
  $maxVisibleDist \leftarrow max(visibleDist(fov_i))$;
  $listCam \leftarrow camDistExtract(tr_n, maxVisibleDist)$;
end foreach
foreach C_1 *of* $listCam$ **do**
  foreach $fov_j(C_i)$ **do**
    if $time(fov_j(C_i)) \geq t_1$ *and* $time(fov_j(C_i)) \leq t_2$ **then**
      $geometry_{i,j} \leftarrow build\_polygon(fov_j(C_i))$;
      foreach tr_n *of the query* **do**
        if $geometry_{i,j}$ *intersect* tr_n **then**
          $addResult(C_i, [time(fov_j), min(succ(time(fov_j)), t_2)])$;
        end if
      end foreach
    end if
    if $time(fov_j(C_i)) < t_1$ *and* $t_1 time(succ(fov_j(C_i)))$ **then**
      $geometry_{i,j} \leftarrow build\_polygon(fov_j(C_i))$;
      foreach tr_n *of the query* **do**
        if $geometry_{i,j}$ *intersect* tr_n **then**
          $addResult(C_i, [t_1, min(succ(time(fov_j)), t_2)])$;
        end if
      end foreach
    end if
  end foreach
end foreach

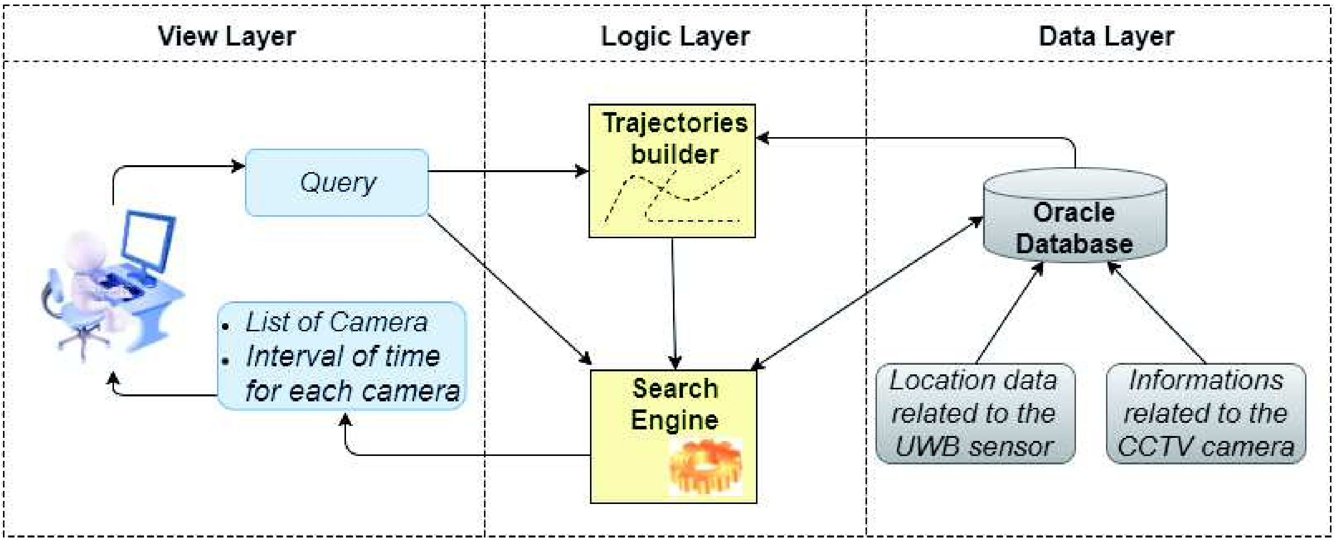

Fig. 9. Architecture of the implementation of our method for automatic retrieval of relevant video content.

3) The "Data" layer: It consists of all the Oracle database specifications as well as those for the collection of location sensor data. The data collection module handles the formatting of location information in well-structured files, which will be saved in Oracles tables through spatio-temporal queries.

B. Implementation

The prototype is implemented using JDK 1.8 with JDBC drivers for the Oracle database connection and JSP technology to manage the connection. The graphical interface is developed using HTML, JavaScript and AJAX. The algorithms implemented are based on two-step techniques (filtering / refining) in order to address the need for rapid response. The main operations are comparisons with linear complexity $\mathcal{O}(n)$. These algorithms run very quickly even with a very large number of operations (e.g. 10,000 operations in 10 milliseconds).

V. EXPERIMENTAL EVALUATION

The datasets come from geolocation sensors installed on the Kyushu University campus in Japan. These datasets consist of 646109 objects detected on the eighth floor from 08/02/14 at 17:29:04 to 19/02/14 at 16:48:28. The scenario consists of 17 CCTV cameras and its description is summarized as an important document theft that occurred on the eighth floor in room 804 on 16/02/14 between 14:05:47 and 17:29:02. The goal is to find elements to identify the suspects.

With our approach, we defined a query whose incident region is the area of the 804 office and the query time interval is [16/02/14 at 14:05:47, 16/02/14 at 17:29:02]. In this scenario, Algorithm 1 detected 13 suspected devices and constructed their trajectories. Algorithm 2 returned 6 cameras with time intervals for each camera, which provided enough information for the investigation. The total number of hours to view for all selected cameras was 00:57:47 (approximately 58 minutes). Manual analysis (without prototype) should consist in watching approximately 3 hours of videos per camera (for 17 cameras), giving a total of 17*3 = 51 hours of video to analyze. Thus, in this scenario, results shows that the developed prototype drastically reduced research space and time for te CCTV operator.

However, since there is no easy way to get the "ground truth" for the query result set, it is difficult to evaluate the accuracy of the matching video segments for the given query. One possible way is the using of object recognition algorithms to extract all video frames in which a given object is visible. However, such object recognition algorithms have their own limitations that would also need to be integrated in the evaluation process. Thus, to evaluate our algorithm, the safest way is to watch all videos and manually set the time intervals in which the desired objects have appeared in each video.

Thereby, we performed experiments in which we defined several queries considering different number of suspects (from 1 to 6). We intersected the trajectories of these devices by the fields of view of the cameras installed in the building. For each query, we performed manual checks of cameras selection and time interval selection, and compared them with our automated approach. Table IV summarized the results obtained. **"Number of selected cameras"** represents all cameras that filmed the suspects during their trajectories; **"Time"** is the total of time intervals for which the selected cameras intersected the suspects trajectories.

Figure 10 shows the number of selected cameras for each query by: manual verification ($Manual\_Check$) and the approach we have proposed ($Proposed\_Approach$).

Results show that, for the most part (5/6), the proposed approach returns the same number of cameras as manual verification. Manual verification shows that the cameras selected by our approach are the correct ones. The only case

TABLE IV
VIDEO RETRIEVAL COMPARISONS.

Number of selected devices (suspects)	Number of selected cameras		Time (minutes)	
	Manual check	Proposed approach	Manual check	Proposed approach
1	2	2	9	10
2	5	5	15	15
3	4	4	18	18
4	3	4	15	17
5	7	7	28	28
6	4	4	19	20

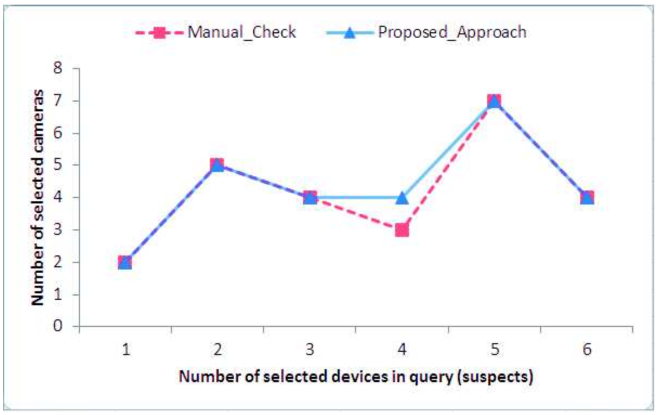

Fig. 10. Number of selected cameras per number of suspects.

where "$Proposed\_Approach$" returns one more camera can be explained by inaccuracy in the generated positions of devices or in camera positions.

Figure 11 shows the total number of times for selected cameras. Results show that, in addition to the time given by $Manual\_Check$, $Proposed\_Approach$ adds one or two more minutes. This is due to the fact that different cameras can film the same segment at a given time, so this redundancy of information increases the total time.

We computed Precision and Recall to evaluate the degree of accuracy and comprehensiveness of our resulting set. We denoted P_C and R_C, the precision and recall measures related to the cameras retrieved by our method. Besides, we denoted P_T and R_T, the precision and recall measures related to the total time of video segments retrieved by our method. In our context, $P_C < 1$ means that the resulting set contains non-relevant cameras, and $R_C < 1$ implies that some relevant cameras have been ignored. Similarly, $P_T < 1$ means the resulting set contains non-relevant time interval, and $R_T < 1$ implies that some relevant cameras have been ignored.

We computed P_C, R_C, P_T and R_T as follows:

$$P_C = \frac{|PA_C(i) \cap MC_C(i)|}{|PA_C(i)|} = 0.96,$$

$$R_C = \frac{|PA_C(i) \cap MC_C(i)|}{|MC_C(i)|} = 1,$$

$$P_T = \frac{|PA_T(i) \cap MC_T(i)|}{|PA_T(i)|} = 0.96,$$

$$R_T = \frac{|PA_T(i) \cap MC_T(i)|}{|MC_T(i)|} = 1,$$

with $PA_C(i)$ and $MC_C(i)$ being respectively the sets of cameras retrieved by $Proposed\_Approach$ and by $Manual\_Check$, regarding the number of suspects i. Similarly, where $PA_T(i)$ and $MC_T(i)$ are respectively the total time of video segment retrieved by $Proposed\_Approach$ and by $Manual\_Check$, regarding the number of suspects i.

We can observed that our method $Proposed\_Approach$ almost performed perfect match with the $Manual\_Check$ except for precision. This shows that our algorithm retrieved the totality of the relevant content without adding too much non-relevant content.

We are aware that the completeness of the results from a single dataset may not imply the same for the general case. Further experiments on a large amount of dataset are needed to evaluate the global performances of the proposed method.

VI. CONCLUSION

In the context of ever-expansion of information generated by CCTV systems, reducing research space and time is a frequent and challenging problem in video analysis for investigation purposes. In this paper, we proposed a method to automatically retrieve video segments that could contain relevant information for the CCTV operator (suspects, trajectories, etc.) in different contexts of indoor environments (supermarkets, metro stations, common workspaces, etc.). The proposed approach consists in using data from ILS in combination with metadata from CCTV cameras to rebuild trajectories of mobile objects and retrieve the video segment of cameras intersecting these trajectories. Algorithms running in two steps (filtering / refining) have been proposed and allow to obtain results that are pairs composed of the selected camera and the associated time interval. Experimental results show that our approach is efficient in providing and a rapid response with relevant content to CCTV operators. One of the hypotheses of our work is that the diversity and the large amount of video content do not allow an exhaustive analysis. Therefore, relevant metadata in the context of CCTV

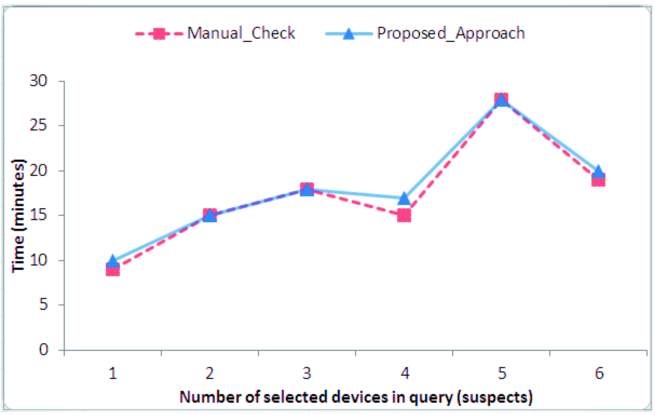

Fig. 11. Total time per query.

must be used to reduce space and implicitly the searching time. This method could also be used to pre-filter irrelevant data for automatic video analyses algorithms, reducing even more the processing-time in real-time investigation cases. In our work, we relied on spatio-temporal metadata to identify videos that are not related to the spatio-temporal trajectory of interest for the investigation. Other "negative" filtering measures can be developed based on metadata or video characteristics in order to improve the information retrieval capability of our approach. We are currently working in collaboration with industry (Thales Communications & Security SA) and technical scientific police (PTS) to ensure that our approach is applicable and effective in the real-world case. The next step of our work is to extend this approach, in FILTER2 French ANR project and VICTORIA H2020 European project.

References

[1] Alarifi A, Al-Salman A, Alsaleh M, Alnafessah A, Al-Hadhrami S, Al-Ammar MA, Al-Khalifa HS. Ultra-wideband indoor positioning technologies: Analysis and recent advances. Sensors. 2016 May 16;16(5):707.

[2] Codreanu, D., Péninou, A., & Sedes, F. Video Spatio-Temporal Filtering Based on Cameras and Target Objects Trajectories–Videosurveillance Forensic Framework. In Availability, Reliability and Security (ARES), 2015 10th International Conference on (pp. 611-617). IEEE.

[3] Codreanu, D., Oria, V., Péninou, A., & Sèdes, F. (2016). Spatio-temporal metadata filtering and synchronising in video surveillance. Ingénierie des Systèmes d'Information, 21(3), 75-91.

[4] Panta, F. J., & Sèdes, F. (2016, October). Querying indoor spatio-temporal data by hybrid trajectories. In Proceedings of the Eighth ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness (pp. 11-18). ACM.

[5] Ilarri, S., Mena, E., and Illarramendi, A. (2010). Location dependent query processing : Where we are and where we are heading. ACM Comput. Surv., 42(3) :12 :1–12 :73.

[6] Afyouni, I., Ray, C., Ilarri, S., and Claramunt, C. (2012b). Algorithms for continuous location-dependent and context-aware queries in indoor environments. In Proceedings of the 20th International Conference on Advances in Geographic Information Systems, SIGSPATIAL '12, pages 329-338, New York, NY, USA. ACM.

[7] Haidar, A. K., Taniar, D., & Safar, M. (2013). Approximate algorithms for static and continuous range queries in mobile navigation. Computing, 95(10-11), 949-976.

[8] Lim J, Bok K, Yoo J. Processing a continuous range query in mobile P2P network environments. InProceedings of the Sixth International Conference on Emerging Databases: Technologies, Applications, and Theory 2016 Oct 17 (pp. 102-105). ACM.

[9] Zheng, Yu. "Trajectory data mining: an overview." ACM Transactions on Intelligent Systems and Technology (TIST) 6, no. 3 (2015): 29.

[10] Imandoust, S.B. and Bolandraftar, M. "Application of k-nearest neighbor (knn) approach for predicting economic events: Theoretical background." International Journal of Engineering Research and Applications 3, no. 5 (2013): 605-610.

[11] Wan Bejuri, Wan Mohd Yaakob & Mohamad, Mohd. (2016). Wireless LAN/FM radio-based robust mobile indoor positioning: An initial outcome. International Journal of Software Engineering and Its Applications Vol.8, No.2 (2014), pp.313-324

[12] Codreanu, D., Manzat, A. M., & Sedes, F., (2013, August). Mobile objects and sensors within a video surveillance system: Spatio-temporal model and queries. In International Workshop on Information Management in Mobile Applications-IMMoA 2013 (pp. pp-52).

[13] Lin, Xin, Jianliang Xu, and Haibo Hu. "Range-based skyline queries in mobile environments." IEEE Transactions on Knowledge and Data Engineering 25.4 (2013): 835-849.

[14] Xiao, G., Li, K., Li, K., & Zhou, X. (2015). Efficient top-(k, l) range query processing for uncertain data based on multicore architectures. Distributed and Parallel Databases, 33(3), 381-413.

[15] Xuan K., Zhao G., Taniar D., Rahayu W., Safar M., Srinivasan B. (2011) Voronoi-based range and continuous range query processing in mobile databases. J Comput Syst Sci 77(4):637–651.

[16] ISO-22311:2012 (2012) Societal security videosurveillance format for interoperability. Technical report, ISO 2012.

[17] Li, T., Chang, H., Wang, M., Ni, B., Hong, R., Yan, S. (2015). Crowded scene analysis: A survey. IEEE transactions on circuits and systems for video technology, 25(3), 367-386.

[18] Gowsikhaa, D., Abirami, S., Baskaran, R. (2014). Automated human behavior analysis from surveillance videos: a survey. Artificial Intelligence Review, 42(4), 747-765.

[19] Parekh, H. S., Thakore, D. G., Jaliya, U. K. (2014). A survey on object detection and tracking methods. International Journal of Innovative Research in Computer and Communication Engineering, 2(2), 2970-2979.

[20] Bedagkar-Gala, A., Shah, S. K. (2014). A survey of approaches and trends in person re-identification. Image and Vision Computing, 32(4), 270-286.