Spatio-temporal metadata filtering and synchronising in video surveillance

Dana Codreanu, Vincent Oria, André Péninou, Florence Sèdes

Spatio-temporal metadata filtering and synchronising in video-surveillance

Dana Codreanu[1], **Vincent Oria**[2], **André Peninou**[1], **Florence Sèdes**[1]

1. *IRIT, University Paul Sabatier*
   Toulouse, France
   {dana.codreanu, andre.peninou, florence.sedes}@irit.fr

2. *New Jersey Institute of Technology*
   Newark, NJ, USA
   vincent.Oria@njit.edu

ABSTRACT. *This paper presents an ongoing work that aims at assisting video-protection agents in the search for particular video scenes of interest in transit network. The video-protection agent inputs a query in the form of date, time, location and a visual description of the scene. The query processing starts by selecting a set of cameras likely to have filmed the scene followed by an analysis of the video content obtained from these cameras. The main contribution of this paper is the innovative framework that is composed of: 1) a spatio-temporal filtering method based on a spatio-temporal modelling of the transit network and associated cameras, and 2) a content-based retrieval based method on visual features. The presented filtering framework is to be tested on real data acquired within a French National project in partnership with the French Interior Ministry and the French National Police. The project aims at setting up public demonstrators that will be used by researchers and commercials from the video-protection community.*

RÉSUMÉ. *Ce papier présente une contribution dont le cadre applicatif vise à aider des agents de vidéo protection dans la recherche de scènes vidéo d'intérêt dans un réseau de transports. L'agent construit une requête à partir d'une date, d'un repère temporel, d'un emplacement et d'une description visuelle de la scène. Le traitement de la requête commence par la sélection d'un ensemble de caméras susceptibles d'avoir filmé la scène, suivie par une analyse du contenu vidéo obtenu de ces caméras. La contribution principale réside dans le cadre novateur qui est composé de : 1) une méthode de filtrage spatiotemporelle basée sur une modélisation spatiotemporelle du réseau de transports et des caméras associées et 2) une recherche basée contenu à partir de caractéristiques visuelles. Le processus de filtrage a été testé sur des données réelles acquises dans un projet national en partenariat avec le ministère de l'Intérieur et la Police nationale. Le projet s'intègre dans le cadre national d'un démonstrateur mis à disposition des académiques et industriels de la communauté.*

KEYWORDS: *video-protection framework, querying, spatio-temporal filtering.*

MOTS-CLÉS : *vidéoprotection, requêtes, filtrage spatio-temporel.*

1. Introduction

Public and private locations nowadays heavily rely on cameras for surveillance. The number of surveillance cameras in service in public and private areas is increasing (e.g., in train and metro stations, on-board of buses and trains, inside commercial areas, inside enterprises buildings). Some estimations show that there are more than 400,000 cameras in London and that the RATP (Régie autonome des transports parisiens - *Autonomous Operator of Parisian Transports*) surveillance system comprises around 9,000 fixed cameras and 19,000 mobile cameras in Paris.

When needed, the video content must be analysed by human agents that have to spend time watching the videos organized in a matrix called video wall. Several studies have shown the cognitive overload coupled with boredom and fatigue that often lead to errors in addition of the excessive processing time. In that context, the main question is how to assist the human agents in order to better do their work?

For instance, regarding the display of the cameras on such a wall in the next illustration (Figure 1), we can see that no spatial nor temporal organization is provided, that requires an extrapolation for the video agent to put them in "sequence" and keep observing in a consistent way.

Many efforts to develop "intelligent" video-surveillance systems have been witnessed in the past years. The majority of these efforts focused on developing accurate content analysis tools (Cucchiara, 2005) but the exhaustive execution of content analysis is resource intensive and, in addition, it gives poor results because of the heterogeneity of the video content. The main idea we put forward in this paper is to use the metadata from different sources (e.g., sensor generated data, technical characteristics) to pre-filter the video content and implement an "intelligent" content-based retrieval.

When a person (e.g., victim of an aggression) files a complaint, she is asked to describe the elements that could help the human agents to find the relevant video segments. The main elements of such description are: the location, the date and time, the victim's trajectory and some distinguishing signs that could be easily noticed in the video (e.g., clothes colour, logos). Based on the spatial and temporal information and, above all, on their own knowledge concerning the cameras location and characteristics, the surveillance agents select the cameras that could have filmed the victim's trajectory. Then, the filtered content is visualized in order to find the target scenes, objects (or people) and events.

Based on these observations, the contribution of this paper concerns the video filtering and retrieval. We did an analysis of the current query processing mechanism within the video-surveillance systems that highlighted the fact that the entry point of any query is a trajectory reconstituted based on a person's positions and a time interval. These elements are used to select the videos of the cameras that are likely to have filmed the scenery of interest. Consequently, the video retrieval is

treated as a spatio-temporal data modelling problem. In this context, we have proposed:

– a definition of the hybrid trajectory query concept, trajectory that is constituted of geometrical and symbolic segments represented with regards to different reference systems (e.g., geodesic system, road network);

– a multi-layer data model that integrates any available relevant data, for instance from open data, such as the road network, the transportation network, the objects movements (including mobile cameras) and the cameras fields of view changes;

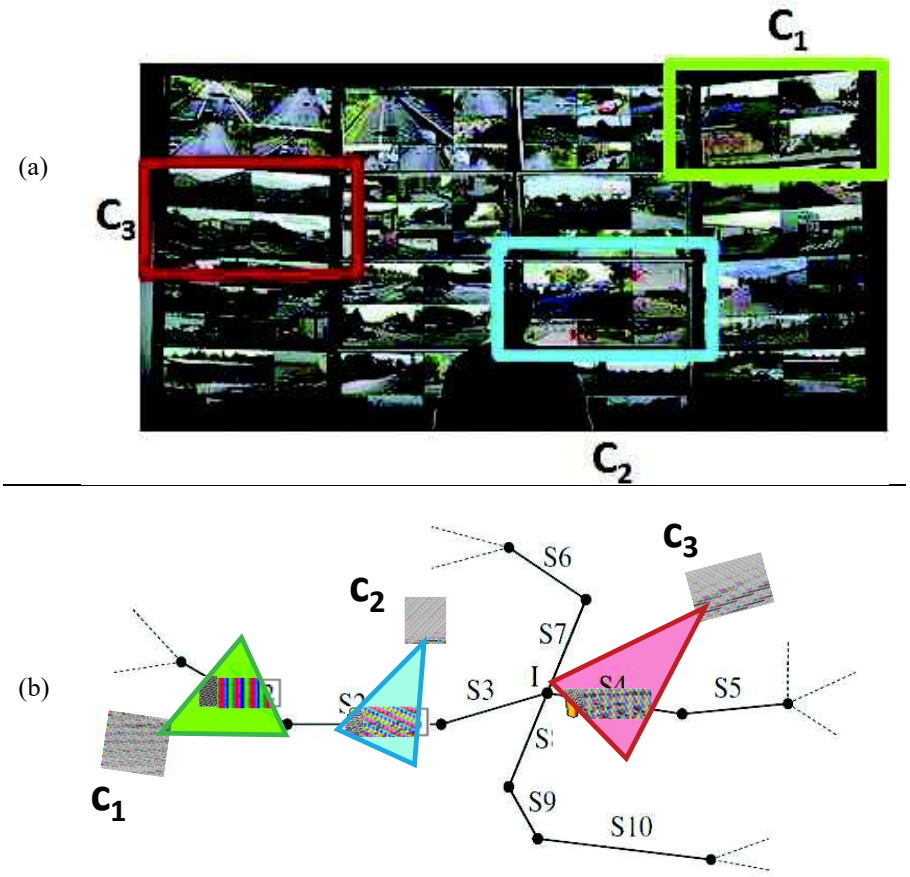

Figure 1. The lack of spatial organization on a video wall. (a) The video wall seen by operators. (b) The real positions of cameras on the road segments

– operators that, given a hybrid trajectory query and a time interval, select the fixed and mobile cameras whose field of view is likely to have filmed the query trajectory.

2. Related works

We present in the following some research projects interested in video retrieval systems focusing on the way they filter the content before executing the feature extractors. The video retrieval projects research generally focuses on developing algorithms based on feature extraction that are exhaustively processed on the available video collections. Very few of them consider a previous video filtering step.

In the following, we present some of these projects with a focus on content filtering before feature extraction. The CANDELA project proposes a generic distributed architecture for video content analysis and retrieval (Merkus *et al.*, 2004). The exhaustive content analysis is done in a distributed manner at acquisition time by a fix set of tools. The CARETAKER project[1] investigates techniques allowing the automatic extraction of relevant semantic metadata from raw multimedia. Nevertheless, there is no filtering of the content before the feature extraction. More related to our work, the VANAHEIM European project[2], based on the human abnormal activity detection algorithms, proposed a technique for automatically filtering (in real time) the videos to display on the video wall screens. Nevertheless, filtering is based on a video analysis relying on a learning process that supposes a big volume of data and that is difficult to implement on a larger scale. The french SURTRAIN project focus on insuring the safety of travellers in public transport systems by the implementation of a system of intelligent transport. It is based on a system of recording of videos and sounds inside trains with software components dedicated to the analysis of images and sounds in order to firstly detect and locate shouts, and then to locate and select the camera the closest to the event. When a camera is selected, tools of follow-up and identification of people can be used to monitor the on-going event. Meanwhile on-going events can be managed in SURTRAIN, the project does not apply to the processing of a posteriori requests. Indeed, a posterori processing of video content is too resource intensive.

In the following, we present research works aiming at organizing and retrieving visual content based on spatio-temporal information. (Liu *et al.*, 2009) propose a system (SEVA) that annotates each frame of a video with the camera location, the timestamp and the identifiers of the objects that appear in that frame. The system consists of: 1) a video camera, 2) a digital compass, 3) a positioning system, 4) a wireless radio associated with the camera. The assumption of the authors is that all objects that may be captured on video are equipped with a system that allows them

1. http://cordis.europa.eu/ist/kct/caretaker synopsis.htm
2. http://www.vanaheim-project.eu/

to transmit their location (which will be captured by the wireless radio). From the location of cameras and the location of objects, images (frames of the video) are annotated by the objects they may contain. The necessary strong assumption (all objects are localizable) implies this solution can only be applied in a controlled environment.

In (Shen *et al.*, 2011), an approach similar to SEVA is proposed with the following differences: (1) the objects don't have to transmit their positions and (2) their objects geometry is considered and not only their localization. For each second of the video, two external databases (OpenStreetMaps and GeoDec) are queried in order to extract the objects (e.g., buildings, parks) that are located in the filmed scene. The list of objects is refined by removing objects that are not visible (by calculating a horizontal and vertical visibility). For each object a list of tags is calculated from external resources (e.g., location, keywords, tag extracted from the associated wikipedia page). The system then retrieves video segments from text queries by computing a similarity between the query words and tags associated with each basic video segment. This system is only usable to query and find identifiable objects having some on-line information and doesn't consider spatial queries.

In (Shahabi *et al.*, 2010), authors present a framework to decision aid based on geospatial information. They have established architecture grounded on a database that integrates information from multiple sources (satellite images, maps, GIS datasets, temporal data, video streams) and that is able to answer to spatiotemporal queries. They develop an interface that facilitates good visualization and interaction. Their proposal brings the existing visualization solutions (Google Maps[3], Google Earth[4]) by improving the interaction with the addition of a time scrollbar. The system does not take more into account the geometry of the cameras field of view, but only positions. So it can help to interactively locate cameras but cannot give information about the ability of a camera to shoot or not some scene.

To conclude this overview, let us give (Epshtein *et al.*, 2007) as one of the various works only related to our work in the way that it proposes a framework based on metadata collected from GPS and compass sensors. Based on a region query, associating each frame of the video with the geometry of the viewable scene, such a framework could return the video sequences that have intersected the video query region based on geolocation criteria.

3. Data model

Based on the state of the art and the user-case requirements we must address, we propose a model that integrates different types of information: 1) the road network, 2) the transportation network, and the objects and sensors that move in this

3. https://www.google.fr/maps/preview
4. http://www.google.com/earth/

environment, 3) objects and 4) cameras. The goal of this model is to gather all data concerning the context of video shooting, and that are necessary to filter videos without content analysis. The final goal is to be able to intersect the trajectory of a person (e.g. a victim) and the shooting zones of fixed and mobile cameras. These data are manipulated by the algorithms presented in the next chapter.

Definition 1: A road network is a non-directed graph $G_R = (E, V)$ where $E = \{e_i / e_i=(v_j, v_k)\}$ is a set of road segments and $V=\{v_i\}$ is the set of segments junctions (Liu *et al.*, 2012).

Definition 2: A transportation network $G_T = (E_T, V_T)$ is a non-directed graph where $V_T = v_{ti}$ is the set of bus station and $E_T = e_{ti} / e_{ti}=(v_{tj},v_{tk})$ is a set of transportation network sections.

Definition 3: $MO=\{mo_i\}$ is the set of mobile object. Let $TR(mo_i) = (P, T)$ be the function that extracts the mobile object mo_i trajectory. Let $P=\{position_j(mo_i)\}$ be the list of mobile object mo_i positions, and let $T=\{time_j(mo_i)\}$ be the mobile object mo_i list of timestamps such as at timestamp $time_j(mo_i)$, the mobile object mo_i is at position $position_j(mo_i)$.

Definition 4: $FC = \{fc\}$ is the set of fixed cameras. With fc being a fixed camera, $id(fc) = c_i$ gives the camera id, $position(c_i)$ gives the camera position and $fov(c_i)$ extracts the set of its *field of view* changes. According to (Arslan *et al.*, 2010), a field of view can be calculated from five characteristics: position, angle of view, orientation, the viewing distance and the size of the sensor. The function $fov(c_i)$ extracts such characteristics so that the field of view of the camera can be calculated. Moreover, the function $time(fov_j(c_i))$ gives the time when the field of view change $fov_j(c_i)$ occurs.

Definition 5: $MC = \{mc\}$ is the set of mobile cameras. With mc being a mobile camera, $id(mc)=c_i$ gives the camera id, $mo(c_i) = mo_i \in MO$ extracts the mobile object to which the camera is attached to. We assume that the camera trajectory is the mobile object one: $TR(c_i) = TR(mo(c_i))$.

We define two types of positions: a *geometric* position that is a 2D position relative to the geodesic system (GPS <lat, long> coordinates) and a *symbolic* position relative to the underlying abstract layers of the data model, the road network (classical addresses) and/or the transportation network (name of bus stop for example). We have defined mapping functions that do the connection between the different layers (e.g., compute the geometric position of a bus station or map an geometric object trajectory with regards to the road network).

Based on the data model, we define the operator *hasSeen* that has as input the query defined as a sequence of spatial segments $(u_1, u_2, ... u_n)$ and a time interval $[t_1, t_2]$. The result is a list of cameras likely to have filmed the trajectory of the query with their corresponding time intervals. The specification of the operator is illustrated in Figure 2. In the result of the operator, for each interval of each camera specification of the form c_i: t_{start} -> t_{end}, u_k, (i) c_i is the id of a camera belonging to FC

or MC, (ii) u_k belongs to the list u_1, u_2,..., u_n of the query and is the segment that may be shot by the camera c_i, (iii) t^i_{start} and t^i_{end} are the timestamps of the beginning and the end of the video of the camera c_i that may shoot u_k, (iv) finally, t^i_{start} and t^i_{end} respect the following constraints to comply with the query: $t_1 \leq t^i_{start}$, $t^i_{start} \leq t^i_{end}$, $t^i_{end} \leq t_2$.

$$hasSeen : u_1, u_2, ..., u_n, [t_1, t_2] => \begin{cases} c_1 : t^1_{start} -> t^1_{end}, u_k (1 <= k <= n) \\ c_2 : t^2_{start} -> t^2_{end}, u_k (1 <= k <= n) \\ ... \\ c_m : t^m_{start} -> t^m_{end}, u_k (1 <= k <= n) \end{cases}$$

Figure 2. The specification of the proposed operator hasSeen

4. The proposed video surveillance framework

Figure 3 illustrates the framework we are proposing in two steps: 1) the spatio-temporal filtering (red workflow in Figure 3) and 2) the multimedia querying (green workflow in Figure 3). Let's suppose the query illustrated in Figure 4 as a running example. We will explain the functioning of our framework based on this query.

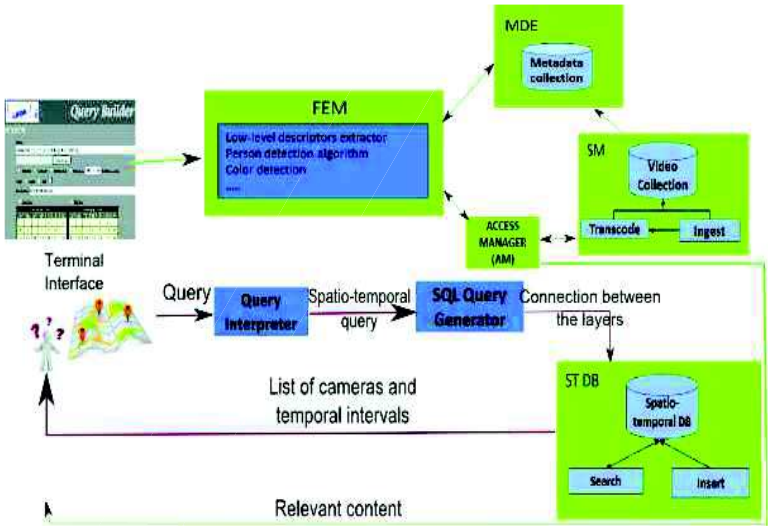

Figure 3. The architecture of the proposed framework

Location : Paris
Date and Time : January 23rd 2014 between 10h and 12h
Trajectory : Rivoli Street : Louvre Museum exit -> Subway Chatelet entrance
Description : man dressed in red

Figure 4. Query example

4.1. Spatio-temporal filtering

Referring to the architecture of Figure 3:

Query Interpreter is the module that "translates" the spatial and temporal information given by the user into a spatio-temporal query.

SQL Query Generator is the module having in input the spatio-temporal query and that implements algorithms 1 and 2.

The main used methods used by the system are:

– *extractCamDist(uk, max(FOV.visibleDistance))* fixed cameras filtering with regards to the query segments and the maximum visible distance of the cameras in the database.

– *Geometries computation and intersection*: compute cameras fields of view geometries and generate SQL queries for intersection with the queries segments; the queries are then executed on *the spatio-temporal database* (data model defined in Section 3).

Spatio-temporal filtering of fixed cameras

Figure 5 illustrates a road network (S1-S5 and S6-S10 are lists *of road segments). The* fixed cameras (C_1, C_2, C_3) positions and fields of view are shown. We suppose the query trajectory is TR = (S_1, S_2, S_3, S_4, S_5) (Rivoli Street: Louvre Museum exit -> Subway Chatelet entrance) and the time interval [t_1, t_2] (January 23rd 2014 between 10h and 12h).

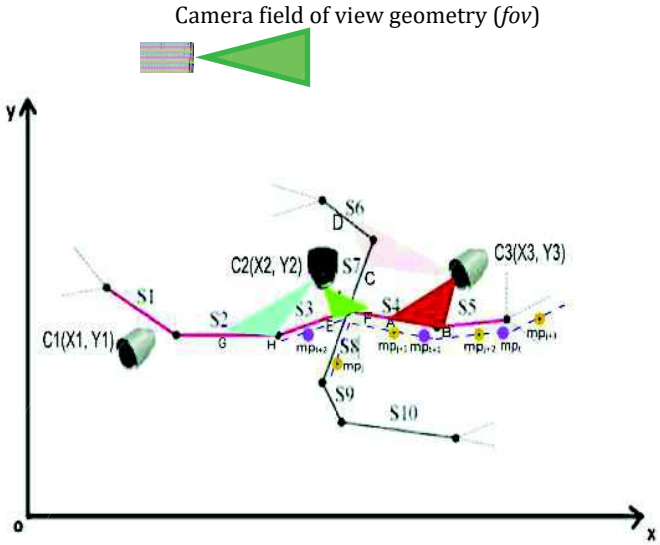

Figure 5. A road network (S1-S5 and S6-S10) filmed by three fixed cameras (C1, C2, C3) and their corresponding field of view

Figure 6 illustrates the different fields of view of the cameras C_2 and C_3 in time (fov(C2) and fov(C_3)). The different times when the fields of view change are marked with colours corresponding to the geometries from Figure 5 (e.g., at time (fov$_j$(C_3)) the field of view becomes AB).

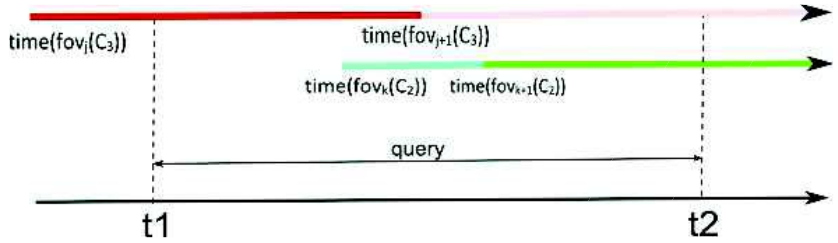

Figure 6. The moments when the fields of view change and the query interval

The algorithm 1 presented hereafter is used to select fixed cameras. The first lines of the Algorithm (1-3) represent a filtering step. From all the cameras in the database we select only those located at a distance smaller than the maximum visible distance from the database. In our case the only cameras that have possibly filmed the query's trajectory segments are C_1, C_2 and C_3.

Algorithm 1: Fixed cameras selection

```
1  for each u_k of the query do
2  |  camList ← extractCamDist(u_k, max(visibleDistance))
3  end
4  for each c_i from camList do
5  |  for each (fov_j(c_i)) do
6  |  |  if time(fov_j(c_i)) ≥ t_1 and time(fov_j(c_i)) ≤ t_2 then
7  |  |  |  geometry_ij ← construct_polygon(fov_j(c_i));
8  |  |  |  for each u_k of the query do
9  |  |  |  |  if geometry_ij intersects u_k then
10 |  |  |  |  |  add(c_i, u_k, [time(fov_j), min(succ(time(fov_j)), t_2)]);
11 |  |  |  |  end
12 |  |  |  end
13 |  |  end
14 |  |  if time(fov_j(c_i)) < t_1 and t_1 ≤ time(succ(fov_j(c_i))) then
15 |  |  |  geometry_ij ← construct_polygon(fov_j(c_i));
16 |  |  |  for each u_k of the query do
17 |  |  |  |  if geometry_ij intersects u_k then
18 |  |  |  |  |  add(c_i, u_k, [t_1, min(time(succ(fov_j)), t_2)]);
19 |  |  |  |  end
20 |  |  |  end
21 |  |  end
22 |  end
23 end
```

For each camera selected at the first step, we then search the periods with changes in the field of view (lines 4-5 of the Algorithm 1). The lines 6-19 process the two possible cases: the change is between t_1 and t_2 (e.g., time($fov_k(C_2)$) or the change is before t_1 (e.g., time($fov_j(C_3)$)). The geometries are built and the intersection with the query's trajectory is evaluated.

As shown in Figure 7, the result of the algorithm for our running example is the following: { (C_2, S_2, [time($fov_k(C_2)$), time($fov_{k+1}(C_2)$)]), (C_2, S_3, [time($fov_{k+1}(C_2)$), t_2]), (C_2, S_4, [time($fov_{k+1}(C_2)$), t_2]), (C_3, S_4, [t1, time($fov_{j+1}(C_3)$)]) }.

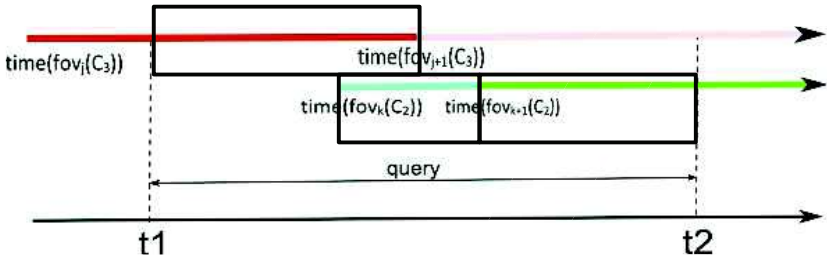

Figure 7. The fixed cameras and the intervals that hasSeen must select (with respect to the query)

Spatio-temporal filtering of mobile cameras

We now consider two mobile objects which trajectories are represented as dotted lines all along the road segments on Figure 5. On the other hand, these trajectories are represented along the time axis in Figure 8.

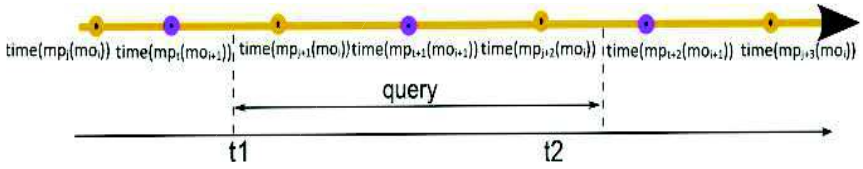

Figure 8. The mobile object trajectory points and the query interval

By mobile object we mean any entity capable of transmitting a periodically update of its position. We assume that each object sends at least one update mp_j (mobile position) per road segment, each mp_j containing its position and a timestamp. We also assume that each mobile object (e.g. a bus) carries at least one camera, which is then a mobile camera. The Algorithm 2 presented hereafter is used to select mobile cameras. By considering each road segment and each mobile object (lines 1-2 of the algorithm 2), the function will test the possible cases: the object's position is on the query's trajectory between t_1 and t_2 (e.g., mp_{t+1}, mp_{j+1}, mp_{j+2} as

illustrated in Figure 8) and the preceding position intersects also (e.g., mp_{j+1} for mp_{j+2}) or the preceding position doesn't intersects the trajectory (e.g., mp_j for mp_{j+1}) or it intersects but before t_1 (e.g., mp_t for mp_{t+1}).

Algorithm 2: Mobile cameras selection

```
for each u_k do
    for each mo_i do
        listMobileObj ← add(filter(mo_i, u_k, [t_1, t_2]));
    end
end
for each mo_i.id from listeObjMobiles do
    listCameras ← selectCamera(mo_i.id);
end
```

As shown in Figure 9, the result of the algorithm for our running example is the following:

{ (obj_i, S_4, [t_1,time(mp_{j+1})]), (obj_i, S_4, [time(mp_{j+1}), time(mp_{j+2})]), (obj_i, S_5, [time(mp_{j+1}), time(mp_{j+2})]), (obj_i, S_5, [time(mp_{j+2}), t2]), (obj_{i+1}, S_5, [time(mp_t), time(mp_{t+1})]), (obj_{i+1}, S_4, [time(mp_t), time(mp_{t+1})]), (obj_{i+1}, S_4, [time(mp_{t+1}), time(mp_{t+2})]), (obj_{i+1}, S_3, [time(mp_{t+1}), time(mp_{t+2})]) }

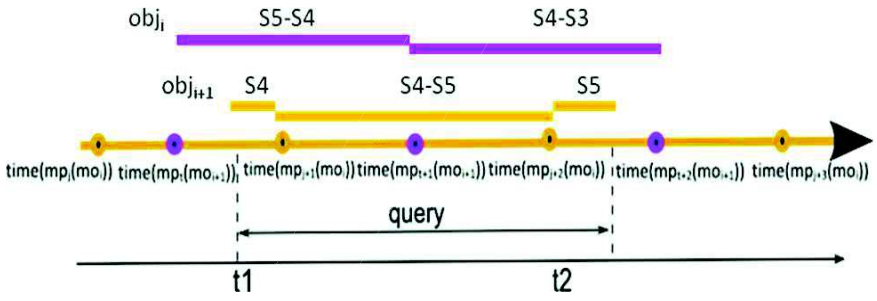

Figure 9. The mobile cameras and the intervals that hasSeen must select (with respect to the query)

4.2. The multimedia retrieval

Once the spatio-temporal filtering is done, the video content is analysed based on the multimedia query engine. Two types of inputs are allowed: 1) textual query (e.g., people dressed in red, etc.) and 2) image query. This search is iterative so for our query example we have the next scenario. The victim remembers that the aggressor was wearing a red coat. The tool that detects people and the main colour of their upper body is processed and the first set of results is presented to the user. He visualizes them and selects a new image query. The image that allowed identification was the one illustrated in the left part of Figure 10.

The LINDO project defined a generic and scalable distributed architecture for multimedia content indexing and retrieval. We used the components of the Video Surveillance server from Paris (described in (Brut *et al.*, 2011)).

Referring to the architecture of Figure 3:

The Access Manager (AM) provides methods for accessing the multimedia contents stored into the Storage Manager (SM). The method the most received from the FEM is *Video extract(String track, long beginTime, long endTime)*: starts the processing of a track between the time beginTime and the time endTime.

The Feature Extractors Manager (FEM) is in charge of managing and executing a set of content analysis tools over the acquired multimedia contents. It can permanently run the tools over all the acquired contents or it can execute them on demand only on certain multimedia contents. The FEM implementation is based on the OSGI framework[5]. The tools or extractors are exported as services and any algorithm that respects the input and output interfaces can be integrated. In our implementation we used tools developed by two of our partners (Supelec[6] and CEA[7] involved with us in several projects) and that are illustrated in Figures 10 and 11.

The Metadata Engine (MDE) collects all extracted metadata about multimedia contents. In the case of a textual query, the metadata can be queried in order to retrieve some desired information. The metadata is stored in an XML format presented in (Brut *et al.*, 2009).

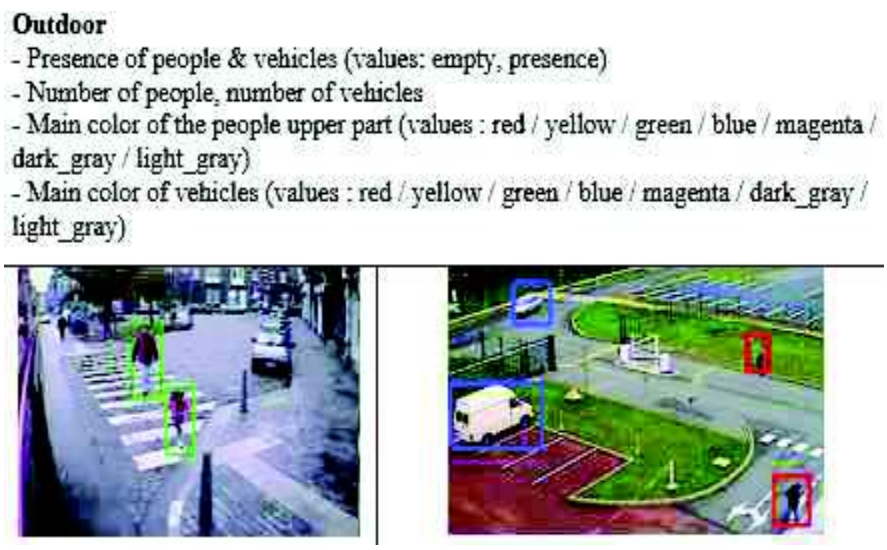

Figure 10. The content analysis tools

5. http://www.osgi.org/Main/HomePage
6. http://www.supelec.fr/
7. http://www-list.cea.fr/

```
<document src="stream1">
 <video capturedBy="cam1_Paris">
  <object type="Person" id="0">
   <localisation confidence="100">
    <period start_time="2010-07-28T11:07:35" end_time="2010-07-28T11:07:55"/>
    <area>parking area entry A2</area>
   </localisation>
   <property name="color">red</property>
  </object>
 </video>
</document>
```

Figure 11. Example of metadata generated by the colour detection tool

5. *A posteriori* validation and use-case

One context of application of our approach is forensic investigations. We have been able to validate the hybrid trajectory algorithm on a use case about the scenario of the "mad gunman attacks" (Paris, Nov. 2013). Given the French Police data, we simulated our prevision of gunman trajectory, and compared it with his pathway. The first step consisted in applying the first algorithm based on fixed camera, with the French National Policy inputs, aggregating different available information layers (subway map, timetable, etc.). In the second one, we introduced the moving devices (bus, cameras, etc.) and generated the hybrid trajectories as shown here after in Figure 12.

Figure 12. The "mad gunman attacks" trajectory and enquiry images

From the different locations where the man has been seen, the cameras are identified and the potential next cameras seeing him in their fields of view are proposed (see Figure 12).

6. Evaluation and future trends in spatio-temporal modelling

In order to validate the different steps of the algorithm, we proceeded to various evaluations. The first results are essentially quantitative, as shown by Figure 13, with the number of retrieved objects, according to the number of segments, and the better score with Requete2 (integrating mobile devices) vs. Requete1 (only fixed cameras).

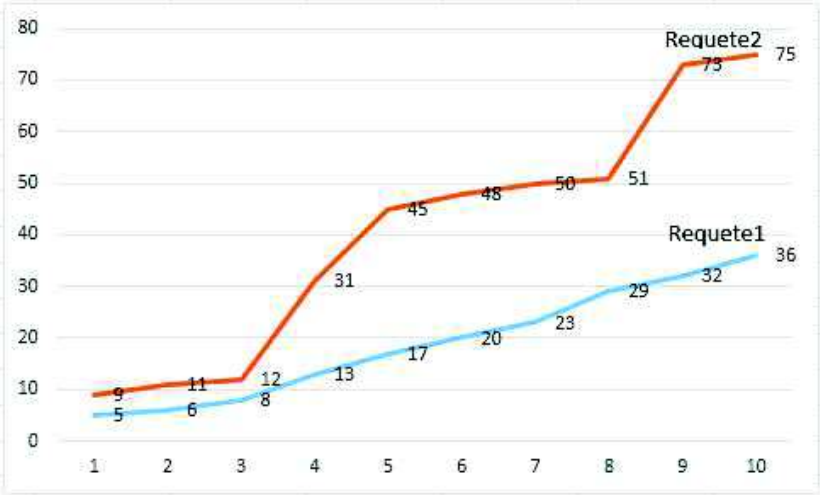

Figure 13. An example of quantitative results (moving camera vs. fixed ones)

Previous works on spatial and temporal uncertainty management in geolocation, teledetection, crisis management, etc. enabled us to keep in mind the need to work with fuzziness in such retrieval process (Alboody *et al.*, 2011). Our contribution to RCC8 spatial reasoning can be very useful in this context (Alboody *et al.*, 2009).

Figure 14 shows the problem of the relevance just before and just after the query timestamps. Figure 15 is about the spatial uncertainty for the field of view of a camera, for instance because of the context (e.g. weather, obstacle, dysfunction, etc.) or its characteristics.

These two aspects of both spatial and temporal uncertainty are critical in order to improve the approach detailed in this paper. On the one hand, the data model and query model are grounded on precise timestamps: timestamps of mobile object positions, timestamps of field of view change of cameras, begin and end timestamps

of query. Nevertheless, real use cases need to “relax” these too precise measures in order to take into account the uncertainty about the values. For example, the begin and end time of interval of the query are subject to imprecision due to perturbation of the victim of an aggression. The timestamps of mobile objects, if they are associated to really precise position in the geodesic system, do not inform about the position of the mobile object “around” the timestamp (before and after). On the other hand, the computation of camera field of view is a well known technique with excellent results in in-lab use-cases. Nevertheless, real use-cases contexts decrease the quality of the results. So, considering some uncertainty in the field of view of camera may increase the possibility to find relevant images for the enquiry and avoid using the maximum visible distance which, by contrast, enlarges too many the results and is not realistic in real use-case (too little discriminating in fact).

Figure 14. Temporal uncertainty when querying

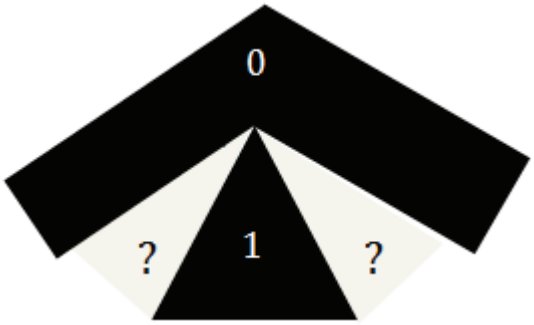

Figure 15. Spatial uncertainty in the field of view of the camera

6. Conclusions

We presented in this paper a video retrieval framework that has two main components: (1) a spatio-temporal filtering module and (2) a content based retrieval

module (based on a generic framework for indexing large scale distributed multimedia contents that we have developed in the LINDO project).

The generic architecture aims to guide the design of systems that could assist the video surveillance operators in their research. Starting from a sequence of trajectory segments and a temporal interval, such system generates the list of cameras that could contain relevant information concerning the query (that 'saw' the query's trajectory) then executes some content analysis tools that could automatically detect objects or events in the video.

To enhance the proposed operators of video retrieval, we plan to take into account fuzzy models of data in order to adjust operators to real use-cases contexts and constraints. For now, our model considers only outdoor transportation and surveillance networks. We are extending our model to indoor spaces also in order to model cameras inside train or subway stations for example, and aggregate trajectories from outdoor/indoor pathways. Finally, we plan to integrate our works on crowdsourcing and social networks (Abascal Mena *et al.*, 2015) to generalize such an approach to social media, adapting the concept of "citizen sensor" (Goodchild, 2007) for instance.

Bibligraphy

Abascal Mena R., Lema R., Sèdes F. (2015). Detecting sociosemantic communities by applying social network analysis in tweets. *In*: *Social Network Analysis and Mining*, Springer, vol. 5, n° 1, december.

Alboody A., Inglada J., Sèdes F. (2009). *Enriching The Spatial Reasoning System RCC8. ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM GIS'08)*, *Irvine, CA, USA*, vol. 1, ACM DL, The SIGSPATIAL Special Number 1, p. 14-20, march.

Alboody A., Sèdes F., Inglada J. (2011). *Enriching The Qualitative Spatial Reasoning System RCC8. Qualitative Spatio-Temporal Representation and Reasoning: Trends and Future Directions*. Shyamanta M. Hazarika (Eds.), Information Science Reference, december.

Arslan Ay S., Kim S. H., Zimmermann R. (2010). Generating synthetic meta-data for georeferenced video management. *In Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, GIS'10,* p. 280-289, New York, NY, USA. ACM.

Brut M., Codreanu D., Dumitrescu S., Manzat A.-M., Sedes F. (2011). A distributed architecture for flexible multimedia management and retrieval. In *Proceedings of the 22nd International Conf. on Database and Expert Systems Applications*, DEXA'11, p. 249-263.

Brut M., Laborie S., Manzat A.-M., Sedes F. (2009). A generic metadata framework for the indexation and the management of distributed multimedia contents. In *3rd International Conf. on New Technologies, Mobility and Security (NTMS)*, p. 1-5, Dec.

Cucchiara R. (2005). Multimedia surveillance systems. In *Proceedings of the Third ACM International Workshop on Video Surveillance and Sensor Networks*, VSSN'05, p. 3-10. ACM.

Epshtein B., Ofek E., Wexler Y., Zhang P. (2007). Hierarchical photo organization using geo-relevance. In *Proceedings of the 15th Annual ACM International Symposium on Advances in Geographic Information Systems*, GIS'07, p. 1-7.

Goodchild Michael F. (2007). Citizens as sensors: the world of volunteered geography, *GeoJournal*, 69, p. 211-221, DOI 10.1007/s10708-007-9111-y.

Liu K., Li Y., He F., Xu J., Ding Z. (2012). Effective map-matching on the most simplified road network. *In Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, p. 609-612.

Liu X., Corner M., Shenoy P. (2009). Seva: Sensor-enhanced video annotation. *ACM Trans. Multimedia Comput. Commun. Appl.*, p. 1-26.

Merkus P., Desurmont X., Jaspers E. G. T., Wijnhoven R. G. J., Caignart O., J. Delaigle J., Favoreel W. (2004). Candela - integrated storage, analysis and distribution of video content for intelligent information systems, *European Workshop on the Integration of Knowledge, Semantics and Digital Media Technology*.

Shahabi C., Banaei-Kashani F., Khoshgozaran A., Nocera, L., Xing S. (2010). Geodec: A framework to visualize and query geospatial data for decision-making. *IEEE Multimedia,* vol. 17, n° 3, p. 14-23.

Shen Z., Arslan Ay S., Kim S. H., Zimmermann R. (2011). Automatic tag generation and ranking for sensor-rich outdoor videos. In *Proceedings of the 19th ACM International Conf. on Multimedia*, MM'11, p. 93-102.