Spatio-temporal metadata filtering and synchronising in video surveillance
Dana Codreanu, Vincent Oria, André Péninou, Florence Sèdes

To cite this version:

HAL Id: hal-01343038
https://hal.archives-ouvertes.fr/hal-01343038
Submitted on 7 Jul 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Open Archive TOULOUSE Archive Ouverte (OATAO)
OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible.

This is an author-deposited version published in: http://oatao.univ-toulouse.fr/
Eprints ID: 15374

The contribution was presented at:
http://bda2015.univ-tln.fr/


Any correspondence concerning this service should be sent to the repository administrator: staff-oatao@listes-diff.inp-toulouse.fr
Spatio-temporal metadata filtering and synchronising in video surveillance

Dana Codreanu  
IRIT, University Paul Sabatier  
Toulouse, France  
dana.codreanu@irit.fr

Vincent Oria  
New Jersey Institute of Technology, NJ, USA  
vincent.oria@njit.edu

André Peninou  
IRIT, University Paul Sabatier  
Toulouse, France  
andre.peninou@irit.fr

Florence Sèdes  
IRIT, University Paul Sabatier  
Toulouse, France  
florence.sedes@irit.fr

ABSTRACT

This paper presents an ongoing work that aims at assisting videoprotection 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 modeling 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.

Categories and Subject Descriptors

H.4 [Information Systems Applications]: Miscellaneous;  
D.2.8 [Software Engineering]: Metrics—complexity measures, performance measures

Keywords

video-protection framework, spatio-temporal filtering

1. INTRODUCTION

Public and private locations nowadays rely heavily on cameras for surveillance and the number of surveillance cameras in service in public and private areas is increasing. But when needed, the content the surveillance videos is analyzed by human agents that have to spend time watching the videos organized in a matrix called video wall. Several studies have showed 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 which tools can assist the human agents better do their work?

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 [3] but the exhaustive execution of content analysis is resource intensive and gives poor results in addition 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 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 color, logos). Based on the spatial and temporal information and on their own knowledge concerning the cameras location, 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 the following elements:

- 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 data of the road network, the transportation network, the objects movement and the cameras' fields of view changes;

• A set of operators that, given a trajectory query and a time interval, select the fixed and mobile cameras whose fields of view are likely to have filmed the query trajectory.

2. RELATED WORK

The video retrieval projects research projects generally focus on developing algorithms based on feature extraction that are exhaustively executed 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 [7]. The exhaustive content analysis is conducted in a distributed manner at data acquisition using a number of tools. The CARETAKER project 1 investigates techniques allowing 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 filter (in real time) the videos to display on the video wall screens. Nevertheless, the filtering is based on a video analysis based learning process that supposes the utilization of a big volume of data and that is difficult to implement on a larger scale.

In the following, we present research works aiming at organizing and retrieving visual content based on spatio-temporal information.

[6], proposes 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. Therefore this solution can only be applied in a controlled environment. In [8], 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 localisation. 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 system doesn’t consider spatial queries. [4] is more related to our work and proposes a framework that associates each frame of the video with the geometry of the viewable scene based on metadata collected from GPS and compass sensors. Based on a region query, the framework can return the video sequences that have intersected the video query region. The main difference between their framework and ours is that they don’t address the multimedia retrieval process.

3. DATA MODEL

We proposed 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 environment (3) Objects and (4) Cameras.

4. THE PROPOSED VIDEOSURVEILLANCE FRAMEWORK

The Figure 3 illustrates the framework we are proposing in two steps: (1) the spatio-temporal filtering (red workflow in the Figure 3) and (2) the multimedia querying (green workflow in the Figure 3). Let’s use the query illustrated in 2 as a running example.

4.1 Spatio-temporal filtering

Query Interpreter is the module that is “translating” the spatial and temporal information given by the user into a spatio-temporal query.

Figure 1: The specification of the proposed operator

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 [5].

Definition 2: A transportation network $G_T = (E_T, V_T)$ is a non directed graph where $V_T = v_i$ is the set of bus station and $E_T = e_{i_1} / e_{i_2}=(v_{j_1}, v_{j_2})$ is a set of transportation network sections.

Definition 3: Let MO= ($m_o_i$) be the set of mobile object. Let TR($m_o_i$) be the function that extracts the mobile object’s $m_o$ trajectory. Let {position$_i$(m$_o_i$)} be the set of mobile object’s $m_o$ positions. Let {time$_i$(m$_o_i$)} be the mobile object’s $m_o$ set of timestamps.

Definition 4: Let FC= {fc$_j$} / fc$_j$ is a fixed camera, id(fc$_j$)= ci, gives the camera’s id, position(fc$_j$)= ci gives the camera’s position and fov(fc$_j$) extracts the set of it’s field of view changes.

Definition 5: Let MC = {mc$_k$} / mc$_k$ is a mobile camera, id(mc$_k$)= ci, gives the camera’s id, mo(mc$_k$)= m$_o$k, EMO extracts the mobile object that the camera is attached to. The camera’s trajectory will be the mobile object’s one: TR(ci$=$ TR(mo$=$ ci)).

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 layers. We have defined mapping functions that do the connection between the different layers (e.g., compute the position of a bus station or map an object’s 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 query’s trajectory with the corresponding time intervals. The specification of the operator is illustrated in figure 1.

Figure 2: Query example
spatio-temporal query implements the algorithms 1 and 2. Here are the used methods:

\[ \text{extractCamDist}(u_k, \max(\text{FOV}.\text{visibleDistance})) \]

fixed cameras filtering with regards to the query segments and the maximum visible distance of the cameras in the database.

generates query segments; the queries are then executed on the Spatio-temporal database.

The fixed cameras (C1, C2, C3) positions and fields of view are shown. Suppose the query trajectory is TR = S1, S2, S3, S4, S5, S10 (Rivoli Street: Louvre Museum exit -> Subway Chatelet entrance) and the time interval [t1, t2] (January 23rd 2014 between 10h and 12h).

The schema from Figure 4 illustrates a road network (S1-S5 and S6-S10). The fixed cameras (C1, C2, C3) positions and fields of view are shown. Suppose the query trajectory is TR = S1, S2, S3, S4, S5, S10 (Rivoli Street: Louvre Museum exit -> Subway Chatelet entrance) and the time interval [t1, t2] (January 23rd 2014 between 10h and 12h).

The Figure 5 illustrates the different fields of view of the cameras C2 and C3 in time (fov(C2) and fov(C3)). The different moments when the fields of view change are marked with colors corresponding to the geometries from the Figure 4 (e.g., at time\(_j\)(fov(C3)) the field of view becomes ABC3).

**Algorithm 1**: Fixed cameras selection

```plaintext
1. for each u_k of the query do
2.    camList ← extractCamDist(u_k, max(\text{visibleDistance}))
3. end
4. for each c_i from camList do
5.    for each (fov(c_i)) do
6.      if time(fov(c_i)) ≥ t_1 and time(fov(c_i)) ≤ t_2 then
7.        geometry\_ij ← \text{construct\_polygon}(fov(c_i));
8.        for each u_k of the query do
9.          if geometry\_ij intersects u_k then
10.         add(c, u_k, [time(fov\_ij), min(time(\text{succ(fov\_ij)}), t_2)]);
11.        end
12.      end
13.    end
14. end
15. if time(fov\_ij) < t_1 and t_1 ≤ time(\text{succ(fov\_ij)}) then
16.    geometry\_ij ← \text{construct\_polygon}(fov\_ij);
17.    for each u_k of the query do
18.      if geometry\_ij intersects u_k then
19.        add(c, u_k, [t_1, min(time(\text{succ(fov\_ij)}), t_2)]);
20.      end
21.  end
22. end
```

The first lines of the algorithm 1 (1-3) represent a filtering step. From all the cameras in the database we will 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 C1, C2 and C3.

For each camera selected at the first step, we will 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\_i(C2)) or the change is before \(t_1\) (e.g., time(fov\_i(C3))). The geometries are built and the intersection with the query’s trajectory is evaluated.

The result is: \{ (C2, S2, [time(fov\_2(C2)), time(fov\_i+1(C2))]), (C2, S3, [time(fov\_i+1(C2)), t_2]), (C3, S4, [time(fov\_i+1(C2)), t_2]), (C3, S4, [t_1, time(fov\_i+1(C3))]) \}.
We now consider two mobile objects which trajectories are represented as dotted lines on the figure 4. By mobile object we understand any entity capable of transmitting a periodically update of its position. Let suppose that each object sends at least one update $mp_i (mobile\ position) containing its position and a timestamp per road segment. By considering each road segment and each mobile object (lines 1-2 of the algorithm 2), the function $\text{filter}(mo, u_k, [t_1, t_2])$ will test the possible cases: the object’s position is on the query’s trajectory between $t_1$ and $t_2$ (e.g., $mp_i, mp_p, mp_{p+1}, mp_{p+2}$ like illustrated in Figure 6) and the preceding position intersects also ($mp_{p+1}$ and $mp_{p+2}$) or the preceding position doesn’t intersects the trajectory ($mp_p$ and $mp_{p+1}$) or it intersects but before $t_1$ ($mp_p$ and $mp_{p+1}$).

The result is: $\{(obj_j, S_1, [t_1, time(mp_{p+1})]), (obj_j, S_5, [time(mp_{p+1}), t_2]), (obj_j+1, S_4, [time(mp_p), t_2])\}$

Figure 6: The mobile object’s trajectory points and the query interval

Algorithm 2: Mobile cameras trajectory selection

```plaintext
def selectionnerCameras(mo, id):
for each $u_k$ do
    for each $mo_i$ do
        listMobileObj ← add($\text{filter}(mo_i, u_k, [t_1, t_2])$); end
end
for each $mo$, $id$ from listeObjMobiles do
    listCameras ← selectionnerCameras(mo, id);
end
```

4.2 The multimedia retrieval

Once the spatio-temporal filtering is done, the video content is analyzed 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 color of their upper body is executed and the first set of results is presented to the user. He watches them and selects a new image query. The image that allowed identification was the one illustrated in the left part of the figure 8.

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 [1]).

The Access Manager (AM) provides methods for accessing the multimedia contents stored into the Storage Manager. The method the most received from the FEM is $\text{extract}($String $\text{track}, long\ beginTime, long\ endTime)$. The FEM implementation is based on the OSGI framework $^3$, 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 the project’s partners $^4$, $^5$ and that are illustrated in figure 8.

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 [2].

5. 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

$^3$http://www.osgi.org/Main/HomePage
$^4$http://www.supelec.fr/
$^5$http://www-list.cea.fr/
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.

For now, our model considers only outdoor transportation and surveillance networks. We plan to extend our model to indoor spaces also in order to model cameras inside train or subway stations for example.

6. REFERENCES


