Surveillance Video Indexing and Retrieval using Object Features and semantic Events

Thi Lan Le, Monique Thonnat, Alain Boucher, François Bremond

SURVEILLANCE VIDEO INDEXING AND RETRIEVAL USING OBJECT FEATURES AND SEMANTIC EVENTS

Thi-Lan Le
MICA, Hanoi university of Technology, Hanoi, VietNam
PULSAR team, INRIA, 2004 route des Lucioles, B.P. 93, 06902 Sophia Antipolis, France.
Lan.Le_Thi@sophia.inria.fr

Monique Thonnat
PULSAR team, INRIA, 2004 route des Lucioles, B.P. 93, 06902 Sophia Antipolis, France
Monique.Thonnat@sophia.inria.fr

Alain Boucher
Equipe MSI, Institut de la Francophonie pour l'Informatique, Hanoi, VietNam
Alain.Boucher@auf.org

François Brémond
PULSAR team, INRIA, 2004 route des Lucioles, B.P. 93, 06902 Sophia Antipolis, France
Francois.Bremond@sophia.inria.fr

In this paper, we propose an approach for surveillance video indexing and retrieval. The objective of this approach is to answer five main challenges we have met in this domain: (1) the lack of means for finding data from the indexed databases, (2) the lack of approaches working at different abstraction levels, (3) imprecise indexing, (4) incomplete indexing, (5) the lack of user-centered search. We propose a new data model containing two main types of extracted video contents: physical objects and events. Based on this data model we present a new rich and flexible query language. This language works at different abstraction levels, provides both exact and approximate matching and takes into account users' interest. In order to work with the imprecise indexing, two new methods respectively for object representation and object matching are proposed. Videos from two projects which have been partially indexed are used to validate the proposed approach. We have analyzed both query language usage and retrieval results. The obtained retrieval results are analyzed by the average normalized ranks are promising. The retrieval results at the object level are compared with another state of the art approach.

Keywords: Surveillance video retrieval, query language, imprecise and incomplete indexing.

1. Introduction

The increasing number of cameras provides a huge amount of video data. Associating to these video data retrieval facilities become very useful for many purposes and many kinds of staff. While some approaches have been proposed for video retrieval in meetings, movies, broadcast news, and sports [37], very few work has been done for surveillance video retrieval [11,12,5]. Current achievements on automatic video understanding [27] such as object detection, object tracking and event recognition, though not perfect, are reliable enough to build efficient surveillance video indexing and retrieval systems. Evaluation has been done for various automatic video understanding approaches on common databases such as CAVIAR [a] (Context Aware Vision using Image-based Active Recognition) [22], ETISEO [b] [23]. To solve the surveillance video indexing and retrieval problem, we need to have both a rich indexing and a flexible retrieval process enabling various kinds of user queries. Two approaches are used to enrich the indexed data. The first one adopts data mining techniques in order to discover new information from the indexed data without user's interaction[24]. The second approach enriches the indexed data by taking into account user knowledge from user interaction. This paper focuses on the second approach.

In surveillance video indexing and retrieval, we are facing five main challenges:

- The first challenge is presented in [28]: "make capturing, storing, finding, and using digital media an everyday occurrence in our computing environment". In video surveillance, while there are many products concerning 'capturing' and 'storing' data, very few work focuses on proposing powerful means for 'finding' the data.
- The second challenge is the lack of different abstraction levels (e.g. images, objects and events) where the indexing phase can be performed. We need a retrieval facility that is able to work at these different abstraction levels.
- The third challenge is the incompleteness of the indexing. By definition indexing means to reduce data so to select pertinent information at the low level (images) or the high level (events). The retrieval phase should produce new information from the indexed ones. For example, a complex event defined by user at the retrieval phase can be inferred from simpler recognized events.
- The fourth challenge is the imprecision of the indexing due to errors in index computation such as object detection, object tracking, object classification and event recognition. The retrieval phase must be able to work with imprecise information.
- The fifth challenge is to be able to take into account the variety of users and of user needs.

This paper presents an approach for addressing these issues. Our approach is based on the following hypotheses:

[a] http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1/
[b] http://www-sop.inria.fr/orion/ETISEO/index.htm

- We suppose that the videos are partially indexed thanks to previous work in video analysis and video surveillance, like object tracking and event recognition.
- We suppose that the videos are raw data directly coming from video sensors, and not structured data scoming from a production or post production phase like in movies or TV programs.

We propose in this paper four main contributions.

The first contribution is a **new data model** that consists of two main abstract concepts: objects and events. The data model is general: (1) it is independent of vision algorithms in the video analysis (2) it allows the video analysis at different levels. The video analysis consists of object detection, object tracking, object classification and event recognition. In our approach, the object detection and tracking are mandatory while object classification and event recognition are optional.

The second contribution is a **new rich and flexible query language**. The query language gives a powerful means for retrieving objects and events of interest. The combination of data model and query language allows us to address the second, third and fifth challenges: (1) Indexing and retrieval are done at several abstraction levels. (2) Our approach is able to detect new events that are defined as a combination of recognized events or a set of objects' relations (3) Users can express their own definition of events.

Our third contribution is a **new method for object representation** based on representative blobs. Representative blob detection algorithm selects the most relevant blobs for each object. Thanks to this algorithm, each object has a compact and meaningful representation.

The fourth contribution is a **new method for object matching**. The proposed method compute similarity measure based on EMD (Earth Movers Distance) that does partial matching and takes into account similarity of each pair of blobs. The third and fourth contributions handle imprecise indexing (the fourth challenge).

The organization of the paper is as follows. Section 2 presents related approaches in surveillance video indexing and retrieval. After that, we propose in section 3 a novel approach enabling to address the previously presented challenges. The proposed approach consists of two phases: the indexing phase and the retrieval phase. They are presented respectively in section 4 and in section 5. Finally, the results of the proposed approach with videos coming from two different applications in video surveillance are given in section 6. The first application is the CARETAKER project (Content Analysis and REtrieval Technology to Apply Extraction to massive Recording). The second is the AVITRACK project (Aircraft surroundings, categorised Vehicles & Individuals Tracking for apRon's Activity model interpretation & ChecK). Videos coming from the first application depict human activity in a metro station while those from the second one depict activities of aircraft, vehicles and people on an apron. A comparison of the proposed approach with Calderara et al.[5] is given.

2. Related Work in Surveillance Video Indexing and Retrieval

The state of the art for image and video indexing and retrieval is very large. This section focuses on only related work in surveillance indexing and video retrieval. The readers are suggested to read [15] for an overview of multimedia information retrieval and [26] for a survey of query language and data model.

In [9], Durak et al. have proposed a video data model for surveillance video archives enabling to work at different abstraction levels. The authors have also analyzed query types on surveillance video archives. However, the defined components in the data model do not show how they can handle the incomplete and imprecise indexing.

For surveillance video retrieval, users often want to retrieve objects or/and events of interest. In [38], the authors have presented a method for object retrieval in surveillance applications. Firstly, the objects are detected and tracked by using the Kalman filter. Then, the MPEG-7 descriptors such as dominant colors, edge histograms are averagely computed over object's life time. This approach addresses partially the second challenge. However, it is not effective because the object detection and tracking are not always perfect. The average descriptors can not characterize reliably the objects.

The approach presented by Ma et al. [18,7] allows to correct the errors of the object detection and tracking if they occur in a small number of frames. For this, the authors have extracted the covariance matrix for each blob of objects. One blob is an instance of object determined by the minimum bounding box in the frame in which it is detected. The distance of two blobs based on the covariance matrix is presented. One object can be detected and tracked in a number of frames. In other words, a set of blobs are determined for each object. In order to remove the errors produced by the object detection and tracking, the authors presented a method for representative blob detection based on the agglomerative clustering. After performing the agglomerative clustering on all blobs of an object, clusters contain a small number of elements (outliers) are removed. For the others clusters, one representative blob are defined for each cluster. As results, one object is represented by a set of representative blobs. For the object matching, the Hausdorff distance is then used to compute the distance between two sets of representative blobs. However, the Hausdorff distance is extremely sensitive to outliers. If two sets of points A and B are similar, all the points are perfectly superimposed except only one single point in A which is far from any point in B, then the Hausdorff distance is determined by that point and is large. The work of Ma et al. gives a partial solution for the second and fourth challenges : The retrieval is at the object level and the representative blob detection algorithm removes few outliers of object detection and tracking.

The approach proposed by Calderara et al. [5] focuses on searching blobs of an object over a network of overlapping cameras. The authors defined PA (person appearance), (SCAT Single Camera Appearance Trace) and MCAT (Multicamera Camera Appearance Trace) for a blob, all blobs of an object observed by a camera and by a network of overlapping cameras respectively. They have proposed a consistent labeling method that connects all SCATs of an object observed by a network of cameras into a MCAT. Typical query is carried out by means of example images (PA). This approach proposes to exploit a two-step retrieval process that merges similarity-based retrieval with multicamera tracking-based retrieval results. The tracking based retrieval finds in MCAT corresponding a PA whose size and the color variation are the biggest. This PA becomes an intermediate query, ten modes of the color histogram are defined for this PA. A mixture of Gaussians is used to summarize a MCAT of an object. The similarity-based retrieval computes the similarity of two mixtures of Gaussians. This approach handles a part of the second and fourth challenges. Object are successfully retrieved if the object detection and tracking are reliable. In the other cases, the mixture of Gaussians of objects is not reliably created and updated. Therefore, object are not successfully retrieved.

Stringa et al. [33] have proposed a system for retrieving abandoned objects detected in a subway station. Two kinds of retrieval units are supported in this work. The first retrieval unit is the frame where the abandoned object was detected. This frame contains the person who left the abandoned object. The second retrieval unit is the video shot. A video shot is composed of 24 frames, the last frame is the frame where the abandoned object was detected. Similar abandoned objects can be retrieved using descriptors such as position, shape, compactness, etc. Retrieval capacity is limited to abandoned object retrieval. Foresti et al. [13] have tried to expand the work of [33] by adding more types of events. The work of Stringa et al. and Foresti et al. addresses the second challenges with a restriction of event types.

A surveillance video retrieval system based on object trajectory has been introduced by Hu et al. [12]. Objects in the scene are firstly tracked and then object trajectories are extracted. The spectral algorithm is used to cluster trajectories and to learn object activity models. Several descriptions are added to the activity models such as turn left, low speed. This approach takes partially into account the second and fourth challenges. It allows to retrieve the indexed data by keywords, multi-objects and sketch-based trajectories. The object activity model enables to work at both the low level and the semantic level. However, the semantic level is limited to only few activities. One modification has been proposed in [36], instead of using spectral algorithm the authors have applied HSOM (Hierarchical Self-Organizing Map) to learn object activity models.

The approach of Ghanem et al. [10] is the first work dedicated for addressing the first and third challenges. The authors have presented an extension of Petri net by adding: conditional transitions, hierarchical transitions and tokens with different types of labels. The general idea of this work is to represent a query by a Petri net whose transitions are simple events modeled also by Petri nets. By this way, a complex event can be inferred from recognized simple events. However, the authors have not explained how this approach can address the imprecise indexing. Moreover, the retrieval at the object level is also not available in this approach.

The IBM Smart Video Surveillance system presented in [11] does both video analysis and video retrieval. Video analysis includes different tasks such as object detection, object tracking, object classification, long-term monitoring and movement pattern analysis. Concerning the retrieval, users are not able to define new events from recognized ones. This approach does not consider temporal relations of events and objects.

For the fifth challenge, while relevance feedback and user interaction have been widely studied in text and image retrieval [30], they are relatively new and difficult for surveillance video retrieval because of the dynamic aspect: objects are detected and tracked in a number of frames; events are defined as a set of relations. Therefore, relevance feedback requires to define which part in a result is relevant (or irrelevant). The technique of Meessen et al. [19] has removed the dynamic aspect by extracting keyframes from videos and applying relevance feedback technique on these keyframes. Chen et al. [6] has limited accident event to one sole relation of vehicles' trajectories.

From the analysis of the previous work in surveillance video indexing and retrieval, no approach has addressed successfully the presented five challenges.

3. Overview of the Proposed Surveillance Video Indexing and Retrieval Approach

Figure 1 gives the general architecture of the proposed approach. This approach is based on an external **Video Analysis module** and on two internal phases: an **indexing phase** and a **retrieval phase**. The external Video Analysis module performs tasks such as mobile object detection, mobile object tracking and event recognition. The results of this module are some Recognized Video Contents. These Recognized Video Contents can be physical objects, trajectories, events, scenarios, etc. So far, we are only using the physical objects and the events but the approach can be extended to other types of Recognized Video Contents. The indexing phase takes results from the Video Analysis module as input data. The indexing phase has three main tasks: **feature extraction**, **object representation** and **data indexing**. It performs feature extraction to complete the input data by computing missing features, object representation for mobile objects and data indexing using a data model. The missing features are features that have been proved robust for object matching. However, with the time constraint video analysis module does not extract. Notes that if Recognized Video Contents are events, they can be put directly to data indexing module. For the mobile objects, in order to work with imprecise object detection and tracking and to reduce the stored information, we detect the representative blobs for each objects. The retrieval phase is divided into four main tasks: **query formulation**, **query parsing**, **matching** and **result browsing**. In the query formulation task, in order to make the users feel familiar with the query language, we propose a SVSQL (Surveillance Video Structured Query Language) language. The vocabulary and the syntax are described in the next section. In the query, the users can select a global image as example or a region in an image from

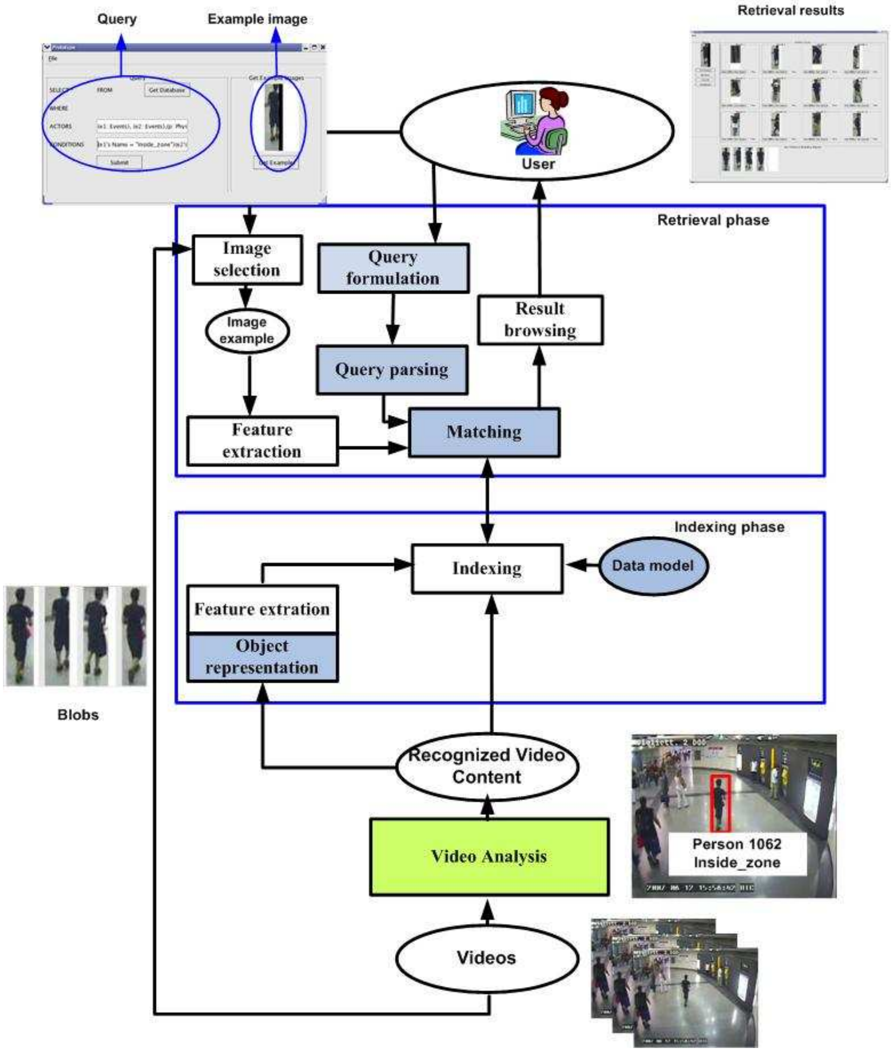


Fig. 1. The global architecture of our approach is consisting of two phases: **indexing phase** and **retrieval phase**. The indexing phase takes results from the **Video Analysis** module as input data and performs **feature extraction**, **object representation** and **data indexing** using a **data model**. The retrieval phase takes queries from users (by the **query formulation** task), analyzes and evaluates them (by the **query parsing** and the **query matching** tasks) using the indexed data from the indexing phase. The retrieval results are ranked and returned to the users (by the **result browsing** tasks). The contributions of this paper concerns the parts in blue.

the database (by the image selection task). In this case, the feature extraction task computes some features in the image example which are used by the query matching task. In the query parsing task, queries built with the proposed language are transmitted to a parser. This parser checks the vocabulary, analyzes the syntax and separates the query into several parts. The matching task searches in the database the elements that satisfy the query. The obtained results are ranked and returned to the users.

Our contributions presented in the introduction of this paper are located into different parts of this architecture. The general data model that allows to make the indexing phase and the retrieval phase at different abstraction levels is determined by the data model based on the results of the Video Analysis module. The representative blob detection algorithm allows to remove the outliers produced by the object detection and tracking and to choose the most relevant blobs for objects. The flexible and rich query language is presented in the query formulation and the query parsing tasks. Both exact matching and approximate matching are provided thanks to powerful features from the feature extraction task and efficient matching in the query matching task. For the approximate matching between two objects, we propose a matching method based on the EMD (Earth Movers Distance) [29]. The user-centered search is done with the aid of the query language.

4. Surveillance Video Indexing

4.1. *Object detection, tracking and event recognition*

There are three key steps in the video analysis: the detection of interesting moving objects (**object detection**), the tracking of such objects from frame to frame (**object tracking**), and the analysis of tracked objects to recognize their behavior (**event recognition**). Every tracking method requires an object detection mechanism either in every frame or when the object appears for the first time in the video. A common approach for object detection is to use the information contained in a single frame. There are four main categories [2] of object detection approaches based on Point detectors, Segmentation, Background Modeling and Supervised Classifiers. The results of the object detection are object regions in the image. It is then the tracker's task to perform object correspondence from one frame to the next one to generate the tracks. In the literature, three main categories for object tracking have been proposed which are: Point Tracking (Deterministic methods and Statistical methods), Kernel Tracking (Template and density based appearance models, Multi-view appearance models) and Silhouette Tracking (Contour evolution, Matching shapes). An overview of object detection and object tracking is given in[2]. Some classifiers such as SVM (Support Vector Machine) are used in order to classify the tracked objects into several predefined classes. Then, the tracked objects are used to recognize events of interest in the video. Two main approaches are used to recognize temporal events from video either based on a probabilistic/neural network or based on a symbolic network. For the computer vision community, a natural approach consists in using a probabilistic/neural network. The nodes of this network correspond usually to events that are recognized at a given instant with a computed probability. For the artificial intelligence community, a natural way to recognize an event is to use a symbolic network whose nodes correspond usually to the symbolic recognition of events. Some artificial intelligence researchers used a declarative representation of events defined as a set of spatio-temporal and logical constraints. Some of them used a traditional constraint resolution or temporal constraint propagation [35] techniques to recognize events.

4.2. *Data model*

Our data model contains two main components: *Physical_objects* and *Events*. The *Physical_objects* are all the objects of the real world in the scene observed by the camera. One physical object belongs to a class and has a list of attributes that characterize this physical object. It can be a contextual object or a mobile object. We are mainly interested in mobile objects. The list of attributes for physical objects is given in Tab. 1. Among these attributes of *Physical_objects*, the ID, Class, 2D_positions, 3D_positions, Time_interval attributes come directly from Video Analysis. The Blobs are representative blobs defined by the object representation task based on all blobs of the object created by Video Analysis. The Features attributes contain the visual features extracted on the representative blobs by the feature extraction task.

Table 1. Attributes of **Physical_objects** in the data model. An attribute written into brackets means that it is optional, i.e. according to applications, it may be used.

Name	Meaning
ID	Label of the physical object
Class	Class that the physical object belongs to.
[Name]	Name of the physical object. It is optional.
2D_positions	List of 2D positions (x, y)
3D_positions	List of 3D positions (x, y, z)
Blobs	List of representative blobs
Time_interval	Time interval in which the physical object appears. It is defined by the starting and the ending frames.
Features	
Histograms	Color histograms
Cov	Covariance matrix
SA	SIFT computed on Shape Adapted regions
MS	SIFT computed on Maximally Stable regions
SA_MS	SIFT computed on both Shape Adapted regions and Maximally Stable regions

In video surveillance, different kinds of states and events can be defined. In order to facilitate the query making, we group them all into one sole 'Events' concept. A list of attributes for Events is defined in Tab. 2.

Table 2. Attributes of **Events** in the data model. An attribute written into brackets means that it is optional. According to applications, it may be used.

Name	Meaning
ID	Label of the event
Name	Name of the event. Ex: Close_to
Confidence_value	Confidence degree of event recognition.
Involved_objects	Physical objects involved in the event.
[Sub_events]	Sub events of the event. It is optional.
Time_interval	Time interval in which the event is recognized

The Blobs and Features attributes aim at doing the approximate matching while the other features are used for the exact matching in the matching task. The Time_interval attribute is used to define the temporal relations of Allen [1] between two objects or two events.

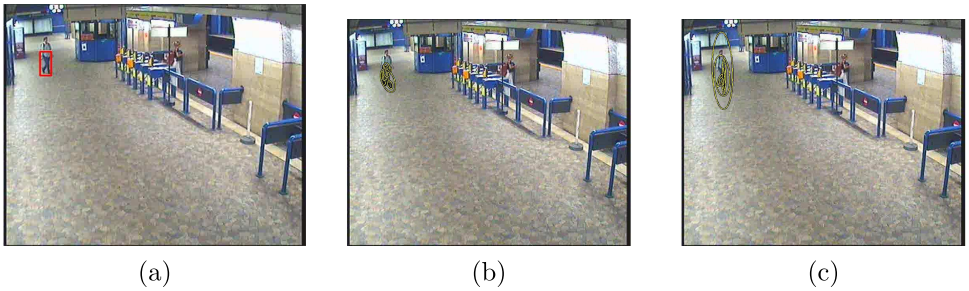
The proposed data model has a good property: the proposed data model is general and independent of any application and of any feature extraction, learning and event recognition algorithm; therefore, we can combine results of different algorithms for feature extraction, learning and event recognition and use them for different application domains.

4.3. *Feature extraction*

The feature extraction task aims to compute low level features describing physical objects; these features are used to make the approximate matching in the retrieval phase. Among the many low level features proposed for image matching, we choose the color histogram [34], covariance matrix [18] and the SIFT descriptor [16] computed over affine covariant regions [20] because they are complementary. The color histogram is a global feature while the SIFT descriptor computed over the affine covariant regions is a local feature. Moreover, the SIFT descriptor computed over the affine covariant regions does not take into account global color information that the color histogram does (note that SIFT descriptor over the affine covariant regions can be computed in both color image [31] and black and white image[25]). However, the SIFT descriptor computed over the affine covariant regions works under affine translation such as rotation, viewpoint. It is a good property for matching images in surveillance video. The color histogram has been presented by Swain and Ballard in [34] and is widely used in image retrieval. An histogram is a vector of certain elements. Each element contains the number of pixels in the image having the color indicated by this element.

Covariance matrix fuses different types of features such as color, texture and relative locations and has small dimensionality. This allows to localize color and texture properties and therefore increase the accuracy of the matching. In this paper, we compute a covariance matrix of 11 × 11 for coordinates of pixels, their color and gradient information.

For computing the SIFT descriptor over the affine covariant regions, we first apply algorithms of affine covariant region detection. Then, we compute the SIFT descriptor over these detected regions. For the affine covariant regions, among the various kinds of affine covariant regions introduced in [20], we are using two types already used in Video Google [32]. The first (Harris Affine) is constructed by elliptical shape adaptation about a shape adapted interest point. The method involves iteratively determining the ellipse center, scale and shape. The scale is determined by the local extremum (across scale) of a Laplacian, and the shape by maximizing intensity gradient isotropy over the elliptical region. This region type is referred to as Shape Adapted (SA). The second type of region (MSER - Maximum Stable Extremal Region) is constructed by selecting areas from an intensity watershed image segmentation. The regions are those for which the area is approximately stationary as the intensity threshold varies. This region type is referred to as Maximally Stable (MS). The two types of regions are employed because they detect different image areas and thus provide complementary representations of a frame. The SA regions tend to be centered on corner like features, and the MS regions correspond to blobs of high contrast with respect to their surroundings such as a dark window on a grey wall. Both types of regions are represented by ellipses. Figure 2 gives an example of Shape Adapted regions and Maximally Stable regions detected in a frame. We choose the SIFT (Scale Invariant Feature Transform) descriptor because the evaluation of local descriptors in [25] shows that the SIFT descriptor proposed by Lowe [16] is superior to others used in the literature such as the response of a set of steerable filters or orthogonal filters. The SIFT algorithm computes a certain number of keypoints for each region. Each keypoint is characterized by its location, scale and orientation and a vector of 128 dimensions.



(a) (b) (c)

Fig. 2. Detected Shape Adapted and Maximally Stable regions from an image of metro station in the CARETAKER project (a) One frame in which a person is detected and tracked, the blob of person being determined by the minimum bounding box (b) Detected Shape Adapted regions in the blob of the person that tend to be centered on corner like features (c) Detected Maximally Stable regions in the blob of the person that correspond to blobs of high contrast with respect to their surroundings.

These features are computed only in the representative blobs of the physical objects. In order to work with various user requirements, we store in index databases five types of features: the color histograms, covariance matrix, the SIFT descriptors computed over the SA regions, the MS regions, and both the SA and the MS regions. At the present, we use these features but it is possible to add new features. According to the applications, new features that are more suited for these applications can be chosen or added.

4.4. *Object representation*

In surveillance applications, one object is in general detected and tracked in a number of frames. In other words, a set of blobs is defined for an object. The number of blobs is huge, the visual difference between two consecutive blobs is small in general. However, there are blobs that are irrelevant for the object due to errors of object detection and tracking. Therefore, the use of all blobs in this set for the object matching is redundant and ineffective. The purpose of the object representation task is to select from a set of N blobs for an object ($B = \{B_i\}$, $i \in 1, N$) created by the Video Analysis module a set of K representative blobs and their weights ($B^r = \{(B^r_j, w^r_j)\}$, $j \in 1, K$) with $K << N$. The representative blobs should be relevant and keep the variety of object's appearance aspect. The associating weight shows the important degree of the representative blob. This information will be used in the object matching. We propose in this paper a new representative blob detection with respect to Ma et al.'s algorithm [18]. As we explain in the state of the art, the algorithm of Ma et al. is able to remove the outliers produced by the object detection and tracking if they occur in a small number of frames. The idea of our algorithm is to remove outliers that occur in a large number of frames using a supervised classification algorithm and to measure the weight of representative blob with respect to the number of blobs it represents. We choose SVM (Support Vector Machine) for supervised classsification. The SVM trained by annotated examples are used to classify blobs of all objects in the videos into relevant and irrelevant blobs. The irrelevant blobs are removed. For each set of blobs of an object, the algorithm performs an agglomerative clustering in order to put blobs in several clusters. Then, it removes the cluster containing few elements and computes the representative blob and its weight for each cluster. The proposed algorithm provides two improvements of Ma et al.'s algorithm: (1) it enables to remove outliers that occur in a large number of frames (2) the weight associated to the representative blob shows the importance degree of this blob in a set of representative blobs. It is used to measure the similarity of two objects. Our algorithm requires to provide annotated examples for training the SVM. However, the SVM is trained one sole times and used for several videos. Figure 3 shows the representation blobs detected by the proposed algorithm. The number of blobs for this person before and after performing the representative blob detection is 905 and 4 respectively.

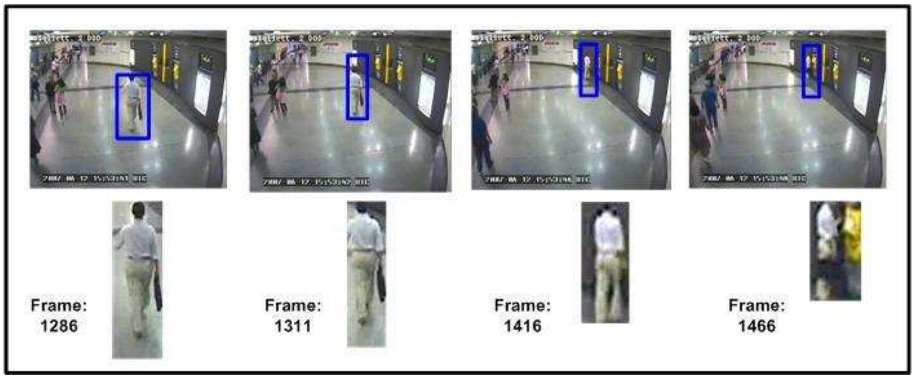


Fig. 3. The representative blobs detected by the representative blob detection algorithm.

5. Surveillance Video Retrieval

5.1. *Query language*

5.1.1. *Query form*

The syntax of a query expressed by SVSQL is the following:

SELECT <Select list > FROM < Database > WHERE <Conditions >

Where: **SELECT**, **FROM**, **WHERE** are keywords for a query and they are mandatory :

- **Select list** specifies the returned results. It may be either whole Physical_objects (or attributes) or whole Events (or attributes). We have implemented an aggregate operator COUNT that counts the number of the returned results.
- **Database** specifies which parts of the video database are used to check the <Condition>. It can be either * for the whole video database or a list of named sub-parts. This is interesting for the surveillance video retrieval because the video database can be divided into several parts according to time or to location. It enables to accelerate the retrieval phase in the case that the users know which parts of the video database they are interested in.
- **Conditions** specifies the conditions that the retrieved results must satisfy. The users express their requirements by defining this component. We distinguish two types of formula in the Condition: a declaration formula (α_d) which is mandatory and a constraint formula (α_c) which provides additional conditions. The declaration formula indicates the types of variable while the constraint formula specifies constraints the variable must satisfy.

A graphic interface can be developed to generate this syntax but it is out of the scope of this paper.

5.1.2. *Conditions*

Declaration formula. The syntax for a declaration formula is: $\alpha_d = (v : type)$ where v is a variable representing the video content to be retrieved. It is there that the user specifies if the retrieval is at the image level, at the semantic level or at both levels. Currently, the two authorized types for the Recognized Video Contents are: *Physical_objects* and *Events*. With subtypes of Physical_objects such as Vehicle, Person, in order to make users feel convenient with the query language, instead of writing (v : Physical_objects) AND (v's Class = subtype of physical objects), we rewrite it as (v : subtype of physical objects). For example: (v : Physical_objects) AND (v's Class = Person) can be rewritten as (v : Person).

In image and video retrieval applications, users usually want to retrieve the indexed data that is similar to an example they have. Therefore, besides the Physical_objects and the Events, we add another type *SubImage*. The SubImage type has Features attribute like the Physical_objects. In a query, $(v : SubImage)$ means that v will be set by user's image example.

Events defined in the queries are detected by event recognition algorithms in the Video Analysis module. The event recognition differs from one application to the other application. We provide a list of recognized events to users so that they can use them to express their queries.

Constraint formula. The syntax for a constraint formula is very rich. We have access functions on attributes of the Physical_objects, the Events and the SubImage as well as temporal and non temporal operators.

The syntax of the access function is $u's\ X$. It means that we access the attribute X of u where u is a variable of the Recognized Video Content defined in the declaration formula. Currently, the authorized access functions are { *'s ID*, *'s Type*, *'s Name*, *'s 2D_positions*, *'s 3D_positions*, *'s Blobs*, *'s Features*, *'s Time_interval*, *'s Histograms*, *'s Cov*, *'s SA*, *'s MS* and *'s SA_MS*} for the Physical_objects and {*'s ID*, *'s Name*, *'s Confidence_value*, *'s Involved_objects*, *'s Sub_events* and *'s Time_interval*} for the Events. These functions are described in Tab. 3. The operator is defined as θ. Besides the comparison operators {=, <, >, >=, =<, ! =}, we use the Allen's interval algebra [1] temporal operators and we propose several non temporal operators based on spatial features. The non temporal operators are listed in Tab. 4.

Based on the access functions and the operators, α_c may have two forms:

- $u's\ X\ \theta\ v's\ Y$. In the special case where this X and Y indicate the same attribute, this form can be rewritten as $u\ \theta\ v$.
- $u's\ X\ \theta\ c$ where c is constant

The constraint formula can be expended by using rules as follows: If α_{c1} and α_{c2}

Table 3. Access functions for attributes of Physical_objects (P), Events (E), and SubImage (I) in SVSQL

Name	Meaning	Components		
		P	E	I
's Id	Get the label	√	√	
's Type	Get the type of physical object	√		
's Name	Get the name	√	√	
's 2D_positions	Get a list of 2D positions	√		
's 3D_positions	Get a list of 3D positions	√		
's Blobs	Get a list of blobs	√		
's Time_interval	Get time interval	√	√	
's Confidence_value	Get the confidence value		√	
's Involved_objects	Get the list of physical objects		√	
's Sub_events	Get the list of sub-events		√	
's Histograms	Get a list of histograms	√		√
's Cov	Get a list of covariance matrix	√		√
's SA	Get Shape Adapted regions + SIFT	√		√
's MS	Get Maximally Stable regions + SIFT	√		√
's SA_MS	Get Shape Adapted + Maximally Stable regions + SIFT	√		√
's Feature	Get all of above features	√		√

are constraint formulae then (α_{c1} AND α_{c2}), (α_{c1} OR α_{c2}) and (NOT α_{c1}) are constraint formulae.

This language is rich enough to enable numerous possible queries. Based on the technique proposed in [35], we implement a parser to check automatically the syntax of the query. This parser automatically analyzes the syntax of the query. The results of this parsing allow to locate which databases will be used to match the query, which variables must be set and which results must be returned.

5.1.3. *Some examples*

An example expressed by this language at the semantic level is: Find Close_to_Gates events occurring in videos from all databases.

*SELECT e FROM * WHERE ((e: Events) AND (e's Name = "Close_to_Gates"))*

where e is a variable of Events, $e's\ Name$ is an access function that get attribute $Name$ of e.

Another example expressed by this language at the image level is: Find the Physical_objects in the database named Video_Database that are similar to a given

Table 4. Non temporal operators on attributes of the Physical_objects (P), Events (E), and SubImage (I) in SVSQL

Name	Meaning	Components		
		P	E	I
color_similarity	Specify if two physical objects or one physical object and a subimage are similar in color	√		√
cov_similarity	Specify if two physical objects or one physical object and a subimage are similar by covariance matrix	√		√
SA_matching	Specify whether two physical objects or one physical object and a subimage are similar by using Shape Adapted regions with SIFTs	√		√
MS_matching	Specify whether two physical objects or one physical object and a subimage are similar by using Maximally Stable regions with SIFTs	√		√
keypoints_matching	Specify whether two physical objects or one physical object and a subimage are similar by using Maximally Stable and Shape Adapted regions with SIFTs	√		√
Features_matching	Specify where two physical objects or one physical object and a subimage are similar by using all of above features	√		√
involved_in	Specify whether a physical object involves an event	√	√	

image.

SELECT p FROM Video_Database WHERE ((p: Physical_objects) AND (i: SubImage) AND (i keypoints_matching p))

where p is a variable of Physical_objects, i is a variable that will be set by an image example, *keypoints_matching* is a non temporal operator.

5.2. *Matching*

Query matching aims at setting the value of the variables in the Declaration formula by using the indexed database indicated in the FROM clause to satisfy the Constraint formula. In our language, for adapting to multimedia databases, we join with each Constraint formula a similarity degree that identifies the level of similarity between the result instance and the query. The returned results are sorted by their similarity degree. The Constraint formula in the query language contains three types of operators (non temporal, temporal and comparison). Our contribution focuses on the non temporal operators.

The actors of the non temporal operators are physical object or subimage. Without lost of generality, we present the non temporal operators for physical objects. The subimage is a special case of physical object (it has one sole blob).

As presented in the object representation section, one object is represented by a set of its representative blobs. Each type of feature extracted on one blob has its own distance such as covariance matrix distances [18] for the covariance matrix. The object matching based on their representative blobs must take into account the similarity of each pair of blobs and their weights. We present a new physical object matching based on the EMD (Earth Movers Distance) [29].

Computing the EMD (Earth Mover's Distance) is based on a solution to the old transportation problem. This is a bipartite network flow problem which can be formalized as the following linear programming problem: Let I be a set of suppliers, J a set of consumers, and c_{ij} the cost to ship a unit of supply from i ∈ I to j ∈ J. We want to find a set of flows f_{ij} that minimize the overall cost:

$$\sum_{i \in I} \sum_{j \in J} f_{ij} c_{ij} \tag{1}$$

subject to the following constraints:

$$f_{ij} \geq 0, \ i \in I, j \in J \tag{2}$$

$$\sum_{i \in I} f_{ij} = y_j, \ j \in J \tag{3}$$

$$\sum_{j \in J} f_{ij} \leq x_i, \ i \in I \tag{4}$$

$$\sum_{j \in J} y_j \leq \sum_{i \in I} x_i \tag{5}$$

where x_i is the total supply of supplier i and y_j is the total capacity of consumer j. Once the transportation problem is solved, and we have found the optimal flow $F^* = \{f^*_{ij}\}$, the earth mover's distance is defined as:

$$EMD = \frac{\sum_{i \in I} \sum_{j \in J} f^*_{ij} c_{ij}}{\sum_{j \in J} y_j} \tag{6}$$

When applied to our problem, the cost c_{ij} becomes the distance of two blobs and the total supply x_i and y_j are the weights of blobs. The c_{ij} is the histogram intersection [34], covariance matrix distance [18], the inverse of number of matched affine covariant regions for the color_similarity, cov_matching, keypoints_matching respectively. As definition, the EMD has some good characteristics: it takes into account the similarity between each pair of blobs and their weights. It allows the partial matching. These characteristics aim at working with imprecise indexing.

The temporal operators are 13 Allen's temporal relations (before, equal, meets, overlaps, during, starts, finishes and their inverses). Each relation verifies a corresponding condition. The similarity is set by 1 (or 0) if the condition is satisfied (or unsatisfied).

For the comparison operators (=, <, >, >=, =<, ! =), the similarity is set by 1 (or 0) if the result of comprison operators is true (or false).

6. Experimental Results and Evaluation

In this section, we first describe the video data that we have chosen to validate the proposed approach. The video data comes from the CARETAKER project. The videos have been automatically processed by the Video Analysis module in order to detect mobile objects and to recognize events. This task has been done by the PULSAR team of INRIA [c]. Then, several experimental retrieval results on the videos of the sole application show that the proposed approach enables to solve the third, fourth and fifth challenges. Finally, we present the query language usage. We show how to use the proposed language to make a query. The proposed language associated with the data model allows to address the first and the second challenges (see the introduction of this paper).

6.1. *Video Databases*

6.1.1. *Input of indexing*

The input of our approach is the Recognized Video Contents coming from the **Video Analysis** module (see Fig. 1). The **Video Analysis** has been done by the PULSAR team. A Video Surveillance Interpretation Platform (VSIP)[4] has been developed by this team. This platform takes a priori knowledge from the application domain and automatically computes object detection, object tracking, object classification and event recognition. Several object classification and event recognition algorithms have been proposed and implemented in this platform. The object classification and the event recognition algorithms are chosen according to the application needs. This platform uses the concepts of "state", "event" and "scenario". A state is a spatio-temporal property valid at a given instant or stable on a time interval. An event is a change of state. A scenario is any combination of states and events. In our approach, we use the "event" concept for state, event and scenario. The event recognition algorithm is presented in [35]. This algorithm uses a declarative language to specify the scenarios. Two examples of state described by the declarative language are given as follows:

*State(**close_to**,*
PhysicalObjects((p : Person), (eq : Equipment))
Constraints((p distance eq $\leq$ *Close_Distance))*

[c] http://www-sop.inria.fr/pulsar/

where Close_Distance is a threshold.

*State(**inside_zone**,*
PhysicalObjects((p : Person), (z : Zone))
Constraints((p in z))

where z is a Zone, *in* is a predicate that checks whether p's center belongs to the polygon z.

The events defined above are recognized for each frame, so the recognized instances which have the same name and involve the same detected person are merged into a sole event. The time interval of this new event is computed as follows. The starting frame of this time interval is the frame containing the earliest recognized instance and the ending frame of this time interval is the frame containing the latest recognized instance. But this event may not be recognized in all the frames included in this time interval. The confidence value is the ratio between the number of frames containing the recognized instances and the total number of frames in the time interval.

6.1.2. *Videos from the CARETAKER project*

The CARETAKER[d] (Content Analysis and REtrieval Technology to Apply Extraction to massive Recording) project aims at studying, developing and assessing multimedia knowledge-based content analysis, knowledge extraction components and metadata management sub-systems in the context of automated situation awareness, diagnosis and decision support. During this project, real testbed sites inside the metro of Roma and Torino, involving more than 30 sensors (20 cameras and 10 microphones) have been provided.

In this project, five classes of physical objects are defined. They are Person, Group, Crowd, Luggage (Lug) and Unknown (Unk). Several primitive states and events are also defined.

We have selected four videos from this project. These videos are acquired by 4 fixed cameras at different positions recording human activities in a metro station. Some example frames extracted from these videos are shown in Fig.4. Each scene contains a platform, several vending machines and gates. Table 6 describes in detail the four videos. In order to facilitate query making, we have named the four videos CARE_1, CARE_2, CARE_3, CARE_4. The four videos have different indexing levels. The object tracking task has been done for the four videos, while event recognition has only been applied for CARE_2.

6.1.3. *Videos from the AVITRACK project*

The AVITRACK [e]project aims at developing an intelligent monitoring system on airport apron, addressing aircraft, vehicles and people movements and actions. The

[d] http://www.ist-caretaker.org/
[e] http://www.avitrack.net/

Table 5. Four videos extracted from the CARETAKER project

Name	Frames	Duration
		(min)
CARE_1	51450	≃ 20
CARE_2	230250	≃ 120
CARE_3	98980	≃ 330
CARE_4	51580	≃ 20

Table 6. The results of video analysis at the object and event levels for these videos

Name	Physical objects					Events	
	Person	Group	Crowd	Lug	Unk	Inside	Close_to
CARE_1	810	-	-	-	-	-	-
CARE_2	29	27	16	25	23	29	19
CARE_3	101	-	-	-	-	-	-
CARE_4	777	-	-	-	-	-	-

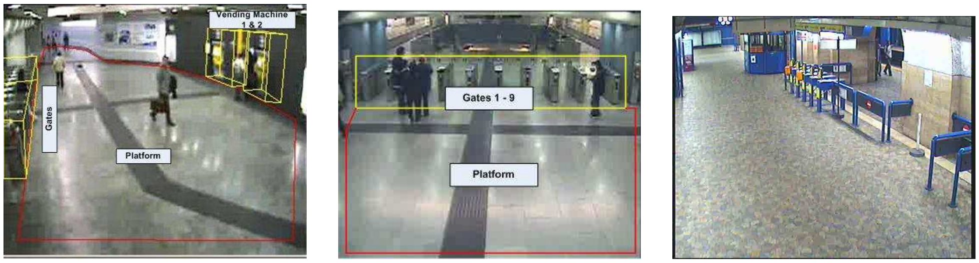


Fig. 4. Some example frames extracted from the videos of the CARETAKER project.

system automatically processes the video sequences and checks timing of movements on the airport apron. It monitors the aircraft parking zone where individuals, objects and vehicles can be detected and tracked.

In this project, expected physical objects are people, ground vehicles, aircraft or equipment. The vehicle type is divided into several sub-types such as GPU (Ground Power Unit) and Transporter. The hierarchical approach that comprises both bottom-up and top-down classification for object recognition has been presented in [14].

Twenty-one generic video events are defined as follows:

- ten primitive states (e.g. a person is located inside a zone of interest, a vehicle is stopped)

- five composite states (e.g. a vehicle is stopped inside a zone of interest: composition of two primitive states stop and located inside a specific zone)
- six primitive events (e.g. a vehicle enters a zone of interest or a person changes from one zone to another)

At present, we use one video from this project. This video consists in 4118 frames, 8 detected persons, 21 detected vehicles. The specific recognized events are "Vehicle_Arrived_In_ERA", Loader_Enters_FrontLoading_Area", "Loader_Departure", "Loader_Stopped_In_FrontLoading_Area", "Loader_Arrival", "Loader_Basic", "Container_Translation", "Loader_Handler_Detected", "Conveyor_Elevation", where ERA is the Entrance Restricted Area. The definition of these events is presented in [14]. Figure 5 presents an example frame of this video.

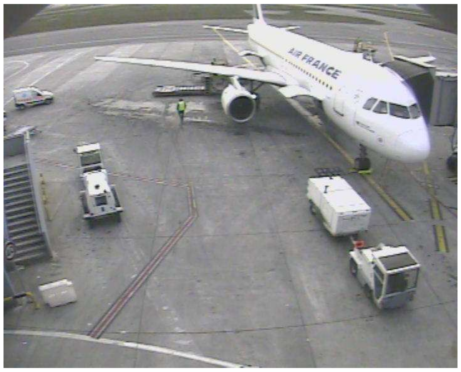


Fig. 5. Example frame extracted from the AVITRACK_1 video

6.2. *Retrieval results*

In this section, we analyze how the proposed approach manages the three challenges: the incomplete indexing, the imprecise indexing and the user-centered search. Objects in videos are not always perfectly detected and tracked. There are three problems (1) one object is detected as several objects (several labels), (2) several objects are detected as one sole object (one sole label), (3) the detected blob of the object does not cover entirely the real object. In [23], the authors have presented criteria to characterize the proportion of the wrong detection and tracking. The quality of event recognition is measured by the number of successfully recognized events. We show how the proposed approach defines new events from the recognized ones.

In order to validate the retrieval results, we adopt the measure of evaluation proposed in [21]: the Average Normalized Rank. It is defined as follows:

$$\widetilde{Rank} = \frac{1}{NN_{rel}}(\sum_{i=1}^{N_{rel}}(R_i) - \frac{N_{rel}(N_{rel}+1)}{2}) \tag{7}$$

where N_{rel} is the number of relevant results for a particular query, N is the size of the tested set, and R_i is the rank of the ith relevant results. $\widetilde{Rank}$ is zero if all N_{rel} are returned first. The $\widetilde{Rank}$ measure is in the range 0 (good retrieval) to 1 (bad retrieval), with 0.5 corresponding to a random retrieval.

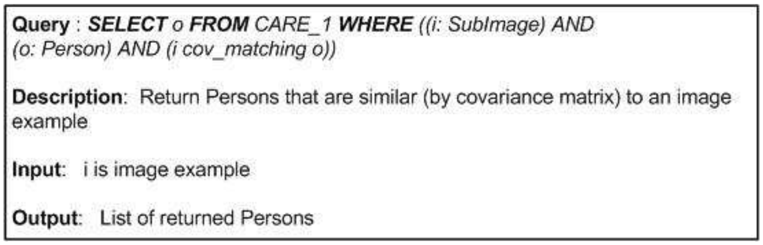
6.2.1. *Working with the imprecise indexing at the object level*

The purpose of this section is to show how the proposed approach handles the imprecise indexing of object. The retrieval results of the proposed approach are compared with Calderara et al. [5]. We have reimplemented the approach of Calderara et al. The parameters of mixture of Gaussians are: $\sigma = 0.1$, $\alpha = 0.01$, initial weight is 0.1. We compare the results obtained by the two approaches in three experiments.

The approach of Calderara performs as the following: in the indexing phase, for an object, a mixture of Gaussians computing the color distribution of all of blobs of the object is created and updated (they do not perform representative blob detection). In the retrieval phase, if the example is an image, the approach of Calderara et al. defines ten modes of color histogram of the example image and matches them with mixtures of Gaussians of indexed objects. If the example is an object, they firstly define the blob of the query object with the largest color variation as an example image.

We summarize our proposed approach: in the indexing phase, we detect the representative blobs and compute the covariance matrices for each object. In the retrieval phase, if the example is an image, the covariance matrix is extracted and compared with the indexed objects using EMD distance. If the example is an object, the covariance matrices are extracted from all representative blobs of this object and compared with the indexed objects using EMD distance.

The first experiment corresponds to the retrieval scenario: "The security staffs have an image example of a person, they want to know whether the same person appears in the scene at different time". The query can be expressed by using the proposed language (see Fig. 6). We have chosen 16 example images. Each example image is compared with 810 indexed persons in the video. Figure 7 shows the results obtained with the proposed approach and that of Calderara et al. for 16 queries of CARE_1. As shown in the Fig. 7, the proposed approach obtains better results in most of queries. The retrieval results of Calderara et al. for the two queries #2 and #8 are better. As we explain in the state of the art, the approach of Calderara et al. obtains good results if the object detection and tracking are reliable (the vision errors occur in a small number or in the first frames). In the other cases (the errors occur in a large number of frames or in the last frames), the mixture of Gaussians is not correctly updated. In these two queries, the persons that are similar to the example image are perfectly detected and tracked in a large number of frames (from 40 to 905 frames). The mixture of Gaussians created for these persons are stable and significant.



Query : ***SELECT*** *o* ***FROM*** *CARE_1* ***WHERE*** *((i: SubImage) AND (o: Person) AND (i cov_matching o))*

Description: Return Persons that are similar (by covariance matrix) to an image example

Input: i is image example

Output: List of returned Persons

Fig. 6. Query 1 retrieves the persons in CARE_1 who are similar to an example image.

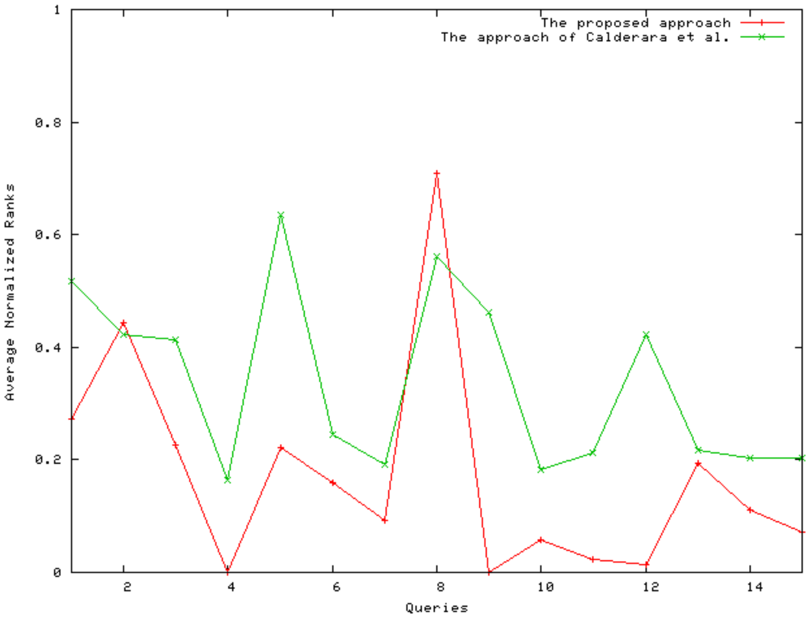
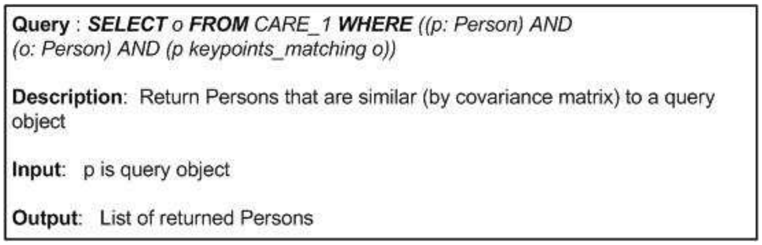


Fig. 7. Ranks obtained with the two approaches for 16 queries over 810 indexed persons of CARE_1.

In the second experiment, the scenario is similar to the first experiment but the query is a person. Figure 8 shows the query expressed by using the proposed language. The results obtained with 247 example persons (Fig. 9) show one more time that the proposed approach is effective for working with imprecise indexing.



Query : ***SELECT*** *o* ***FROM*** *CARE_1* ***WHERE*** *((p: Person) AND (o: Person) AND (p keypoints_matching o))*

Description: Return Persons that are similar (by covariance matrix) to a query object

Input: p is query object

Output: List of returned Persons

Fig. 8. Query 2 retrieves the persons in CARE_2 who are similar to an example person.

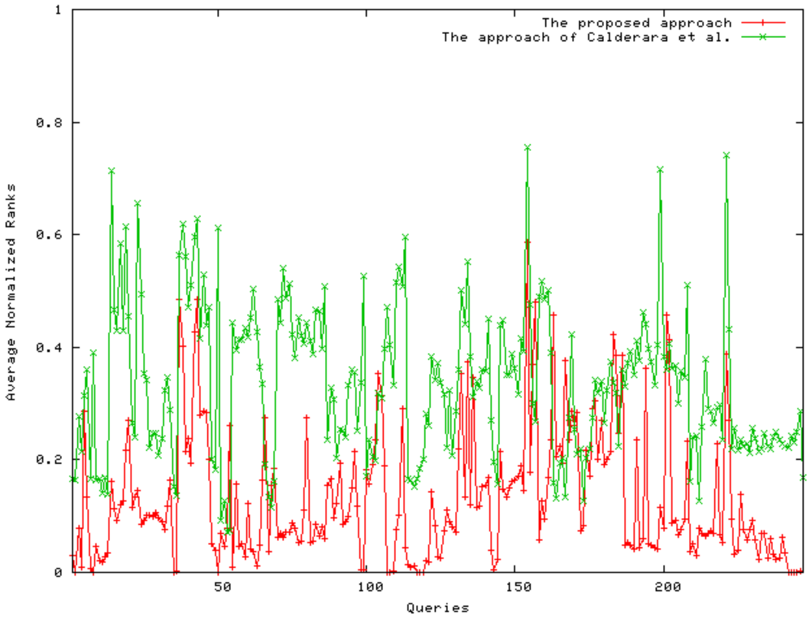


Fig. 9. Ranks obtained with the proposed approach and that of Calderara et al. for 247 query persons over 810 indexed persons of CARE_1.

The retrieval scenario in the third experiment is: “The security staffs know about a person observed by a camera, they want to know whether this person is observed by another camera.” The query is similar to the second experiment. The proposed approach obtains better results in 36 out of 54 queries (see Fig. 10). The reason why our approach does not obtain good results in 18 queries is: the proposed approach uses all of representative blobs of query object while the approach of Calderara et al. takes one sole blob with the largest color variation. The representative blobs are sometime not all relevant for the object due to the errors of the representative blob detection algorithm as shown in Fig. 11. We analyze retrieval results of the query

#6 and #40. Calderara et al.'s approach obtains better results with query #6 but it is not the case with query #40. In query #6 (see Fig. 12), the detection and tracking of relevant person is relatively perfect. The 10 modes of color histogram of query blob are similar to the mixture of Gaussians for relevant persons. Our approach retrieves irrelevant persons due to the dominant presence of vending machines and other persons in blobs. Our approach is effective in query #40 (see Fig. 13). It retrieves the relevant results in the first results. Even for irrelevant results, the retrieved persons are very similar to the query person. In this query, Calderara et al.'s approach shows its limitation. The retrieval persons are not successfully detected and tracked. The blobs of person are their shadows or walls in platform). Moreover, the mixture of Gaussians is based on color distribution, it looses position information.

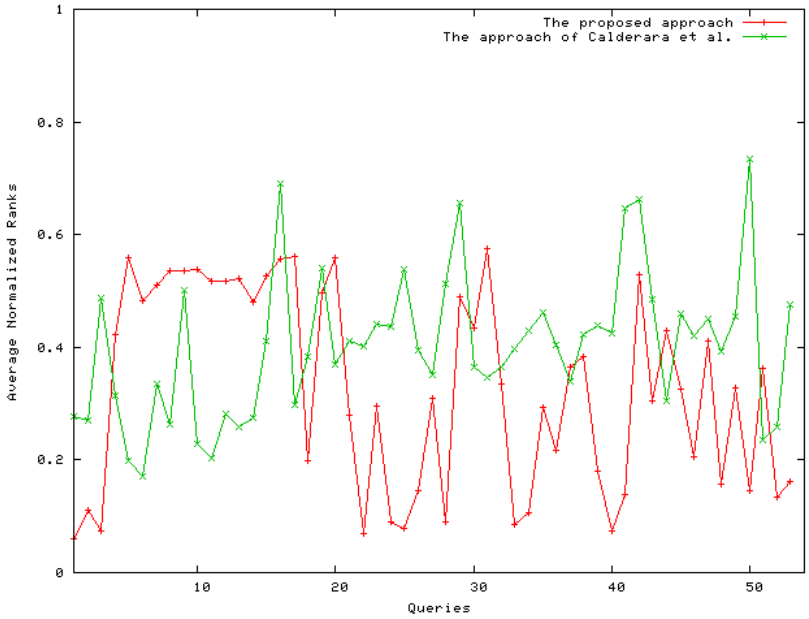


Fig. 10. Ranks obtained with the two approaches for 54 query persons of CARE_4 over 810 indexed persons of CARE_1.

6.2.2. *Working at both the object level and the event level*

The proposed approach enables to retrieve the indexed data at both levels: object and event. Query 3 (Fig. 14) shows an application of this capacity. This query retrieves interesting events with a rich description to enable to refine the favor processing. For instance, the event *close_to_Gate1* (p) indicates that person p is close

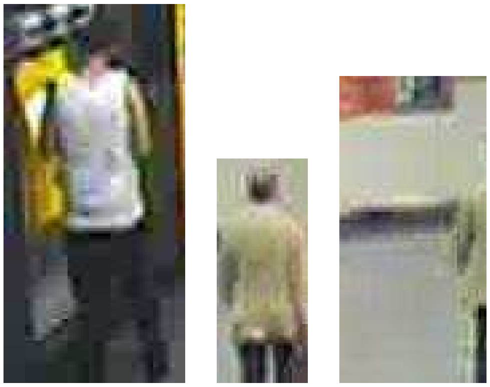


Fig. 11. A label associated with three representative blobs. Several blobs are not relevant for this person.

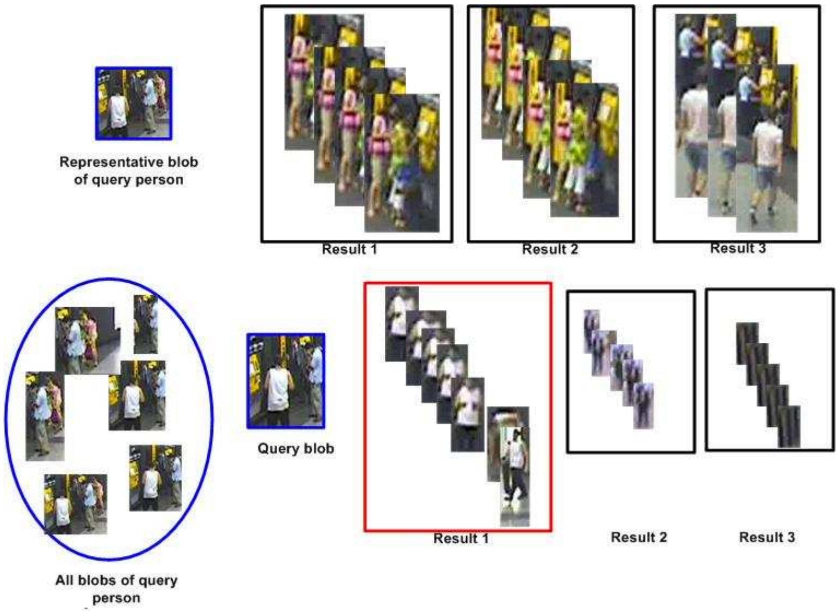

Representative blob of query person
Result 1
Result 2
Result 3
All blobs of query person
Query blob
Result 1
Result 2
Result 3

Fig. 12. Retrieval results for query #6 of the experiment 3. The top images are the representative blobs of the query person and three retrieval results of our approach. The bottom images are all blobs of the query person, the query blob and retrieval results of Calderara's approach. The result in red is relevant result.

to the Gate 1. In the case of many *close_to_Gate1* recognized instances, the user may be interested in only *close_to_Gate1* frames containing a person who is similar to a given example. To answer this query, the persons involved in all *close_to_Gate1*

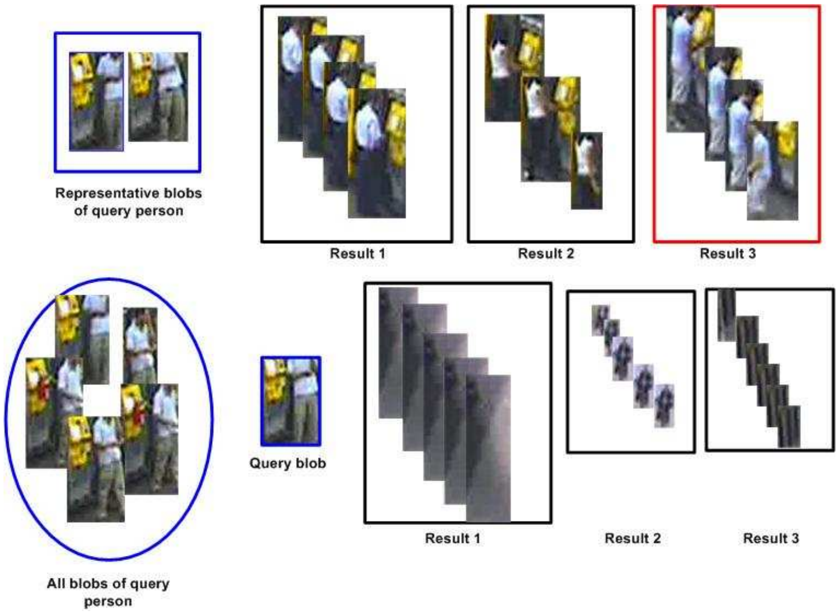
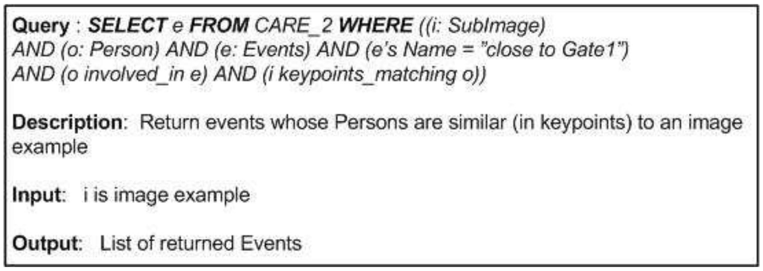

Representative blobs of query person
Result 1
Result 2
Result 3
All blobs of query person
Query blob
Result 1
Result 2
Result 3

Fig. 13. Retrieval results for query #40 of the experiment 3. The top images are the representative blobs of the query person and three retrieval results of our approach. The bottom images are all blobs of the query person, the query blob and retrieval results of Calderara's approach. The result in red is relevant result.



Query : ***SELECT*** *e* ***FROM*** *CARE_2* ***WHERE*** *((i: SubImage)*
AND (o: Person) AND (e: Events) AND (e's Name = "close to Gate1")
AND (o involved_in e) AND (i keypoints_matching o))

Description: Return events whose Persons are similar (in keypoints) to an image example

Input: i is image example

Output: List of returned Events

Fig. 14. Query 3 returns close_to_Gate1 events whose Persons are similar (in keypoints) to an image example.

events are checked for *keypoints_matching* process with the given example. The returned results for each query is a list of *close_to_Gate1* events ranked by the number of matched keypoints between the involved persons and the given example. In 19 *close_to_Gate1* events, there are several events concerning one person. We have chosen 15 example images. Each image turns as input for the query. Figure 15 gives the obtained average normalized rank over 15 image examples and 19 events of *close_to_Gate1* in the CARE_2 video of the CARETAKER project. The ground

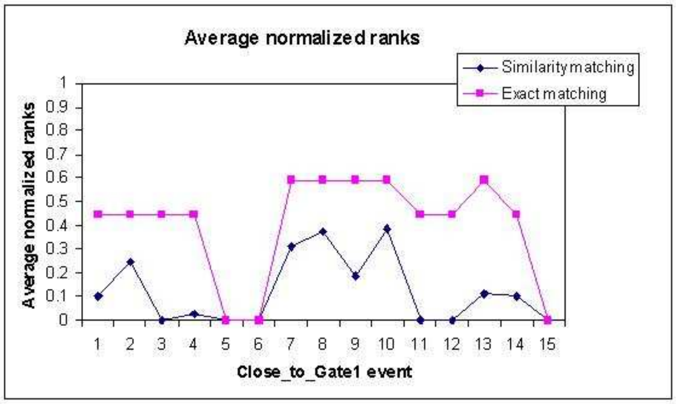
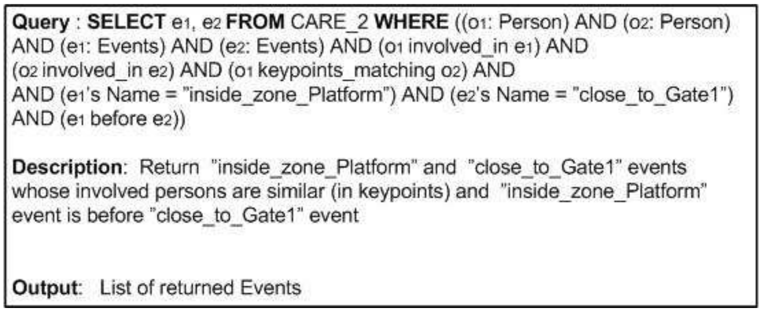


Fig. 15. The average normalized ranks for Query 3 over 15 example images. Each point represents an obtained rank for one value of i. These obtained rank values are much smaller than the rank of random retrieval. The value 0 of average normalized ranks corresponds to good retrieval, value 1 corresponds to bad retrieval and 0.5 corresponds to random retrieval.

truth has been made manually for these 15 queries. A returned result is considered relevant if it is a *close_to_Gate1* event whose the involved persons show the same person as in the given image example. The obtained average normalized ranks are much smaller than the random retrieval. The retrieval result shows that the proposed approach enables users to have queries by combining information of two levels: the object level and the event level.

6.2.3. *Working with the incomplete indexing*


Query : **SELECT** e1, e2 **FROM** CARE_2 **WHERE** ((o1: Person) AND (o2: Person) AND (e1: Events) AND (e2: Events) AND (o1 involved_in e1) AND (o2 involved_in e2) AND (o1 keypoints_matching o2) AND AND (e1's Name = "inside_zone_Platform") AND (e2's Name = "close_to_Gate1") AND (e1 before e2))

Description: Return "inside_zone_Platform" and "close_to_Gate1" events whose involved persons are similar (in keypoints) and "inside_zone_Platform" event is before "close_to_Gate1" event

Output: List of returned Events

Fig. 16. Query 4 retrieves indexed persons who move from Platform to Gate1.

By definition indexing means to reduce data so to select pertinent information at the low level (images) or the high level (events). Another advantage of our approach is to define and to recognize new events from the recognized ones. For example, from two recognized events in the database, inside_zone_Platform and close_to_Gate1, the user wants to retrieve a composite event consisting of inside_zone_Platform and close_to_Gate1 events that satisfy the constraint *before* (first event before the second). Because of imperfect indexing, one person in the real world may be indexed as different persons in the database. An exact matching that matches the indexed persons by their labels would have returned for this query not complete results and sometimes empty ones. Therefore, in this query, user uses the similarity matching between Persons involved in both events to determine if Person involved in inside_zone_Platform event is the person involved in close_to_Gate1 event. The query is expressed in Fig. 16.

In our proposed approach, the system first matches the involved persons in both inside_zone_Platform and close_to_Gate1 by keypoint matching. For each person involved in the close_to_Gate1 event, it computes the number of matched keypoints between this person and the persons involved in the inside_zone_Platform event. A set of persons ordered by their matched keypoints are returned. The inside_zone_Platform events containing these persons become candidates for retrieval results. Then, these events are used to check whether they satisfy the *before* constraint with the close_to_Gate1 event.

For each close_to_Gate1 event, the retrieval result is a list of inside_zone_Platform events that satisfy the *before* constraint with the close_to_Gate1 event and are ranked by the number of matched keypoints between their involved persons and the person involved in the close_to_Gate1 event. We are working with 19 close_to_Gate1 events and 29 inside_zone_Platform events. The obtained average normalized rank is given in Fig.17 over 19 events of close_to_Gate1. The ground truth has been made manually. For each close_to_Gate1 event, the returned result is considered relevant if it contains an inside_zone_Platform event that satisfies the *before* constraint and if their involved persons show the same person in the real world.

This query shows the capacity of this language to define new events from the recognized ones with satisfying results (average normalized ranks of all 19 events are smaller than 0.3).

Figure 18 gives an example in which the results of exact matching are not complete. Three indexed persons with labels 10, 8, 7 describe the same person in the real world. These indexed persons belong to close_to_Gate1 and inside_zone_Platform events that satisfy *before* constraint. The exact matching based on person's label gives only one result of person label 10 while our similarity matching gives three persons label 10, 8, 7 with respectively 1, 2, 5 of rank.

6.2.4. *Taking into account users' interests and users' degrees of knowledge*

The proposed approach is flexible because it adapts itself to the users' needs. We explain it in the following query. In the CARE_3 video of the CARETAKER project, we do not have the results of event recognition. Users may define then

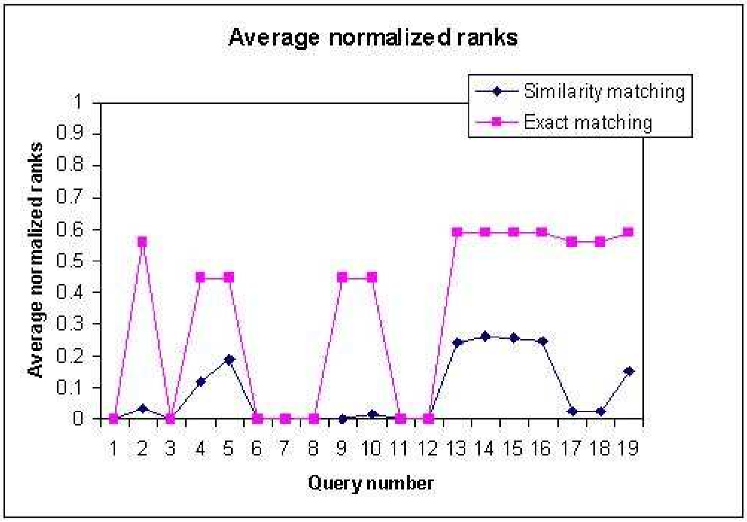


Fig. 17. The average normalized rank for Query 4 over 19 close_to_Gate1 events and 29 inside_zone_Platform events. The value 0 of average normalized ranks corresponds to good retrieval, value 1 corresponds to bad retrieval and 0.5 corresponds to random retrieval.

Close_to_Gates event by stating Query 5 (Fig. 19). One person is close to gates if the distance between this person and gates is smaller a given threshold. The distance is computed based on 3D position of the persons and gates. This query is different from Query 4 because it takes into account user's interest by using a threshold. User can set for the threshold a value as he/she wants. By setting two different values for the threshold, 100 and 150, we have two different results. The first result returns 10 indexed persons with 320 recognized instance of the Close_to_Gates event. The second one returns 20 indexed persons with 727 recognized instances.

6.3. *Query language usage*

Query making in the proposed language can be done at different levels: queries concerning the object level, queries concerning both the object and the event level, queries concerning the event level. In the queries concerning the object level, object features (such as the color histogram, the SIFT descriptors on the Shape Adapted and Maximally Stable regions, trajectory, size, class, position in the scene) can be used to retrieve physical objects from the database. Queries in the second category try to get physical objects and events based on event name, object class, object features, and non temporal and temporal relations between events and physical objects. Queries concerning the event level are based on event characteristics such as event name and temporal relations between events.

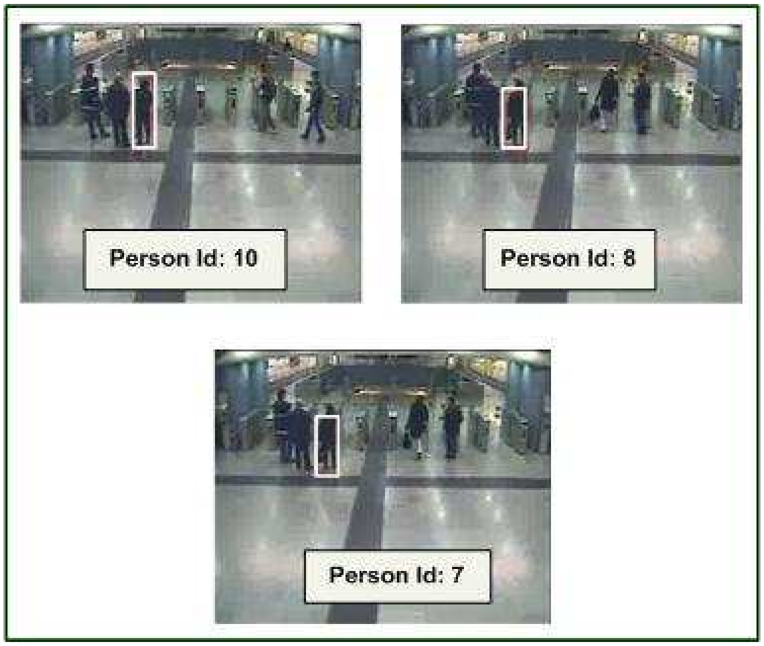
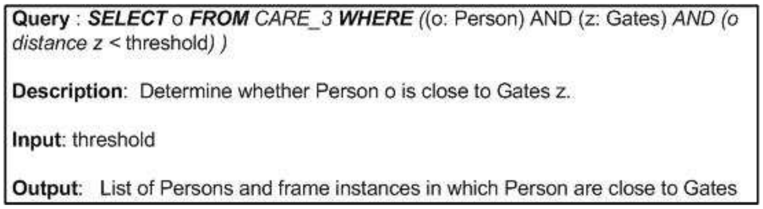


Fig. 18. Three indexed persons with labels 10, 8, 7 describe the same person in the real world. These indexed persons belong to close_to_Gate1 and inside_zone_Platform events that satisfy the *before* constraint. The exact matching based on person's label gives only one result of person label 10 while our similarity matching gives three persons label 10, 8, 7 with respectively 1, 2, 5 of rank.



Query : ***SELECT*** o ***FROM*** *CARE_3* ***WHERE*** *((*o: Person) AND (z: Gates) *AND (o distance z* < threshold*))*

Description: Determine whether Person o is close to Gates z.

Input: threshold

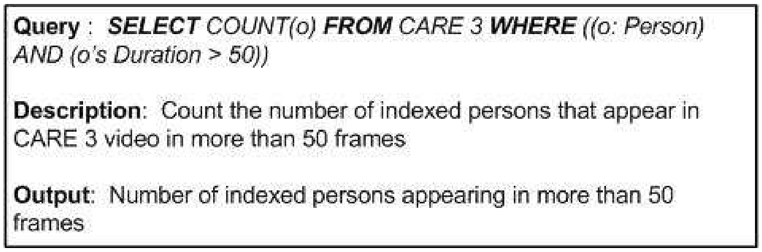
Output: List of Persons and frame instances in which Person are close to Gates

Fig. 19. Query 5 allows users to define their own Close_to_Gates event by setting the value of the threshold.

6.3.1. *Queries at the object level*

This category of query allows to retrieve the indexed objects in the database whose attributes satisfy the given criteria. Criteria are checked based either on the exact matching or the approximate matching according to the chosen attributes. In the retrieval results section, we present several queries at object level using the similarity matching. We describe one more query at this level for exact matching.

Query 6 (see Fig. 20) counts the number of the indexed persons that appear



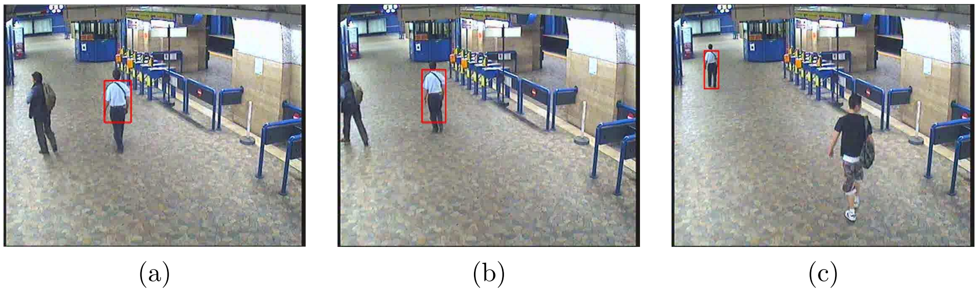
Query : ***SELECT*** *COUNT(o)* ***FROM*** *CARE 3* ***WHERE*** *((o: Person) AND (o's Duration > 50))*

Description: Count the number of indexed persons that appear in CARE 3 video in more than 50 frames

Output: Number of indexed persons appearing in more than 50 frames

Fig. 20. Query 6 counts the number of the indexed persons that appear in the CARE_3 video in more than 50 frames.

in the CARE_3 video in more than 50 frames. Among 1616 indexed persons in the CARE_3 video, 101 indexed persons have a duration between 51 and 1556 frames and are correctly returned. Figure 21 presents one of the 101 indexed persons returned for this query.

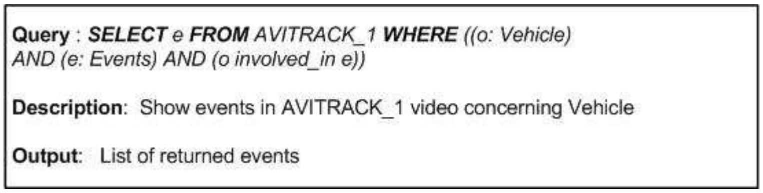


(a) (b) (c)

Fig. 21. One of the 101 indexed persons returned for the query 6: the person with label 122 and duration 56 frames. This person goes along the platform in CARE_3 video (a) from frame 2376 (b) to frame 2381 (c) and to frame 2407.

6.3.2. *Queries at both the object and event levels*

The proposed language allows to have queries containing both the object level and the event level. This is interesting because it can answer two user crucial questions: which events happen in the video and who is involved in these events (Physical_objects). We present two queries in this category. Query 7 (Fig. 22) searches events in AVITRACK_1 concerning an object class (in this case, the object class is Vehicle) and Query 8 (Fig. 23) returns Vehicle involved in a specified event (event name is set "Vehicle_Arrived_In_ERA"). Query 7 returns 7 "Loader Basic", 13 "Vehicle Arrived In ERA", 1 "Loader Departure", 1 "Loader Enters FrontLoading Area", 7 "Container Translation", 10 "Loader Handler Detected", 5 "Conveyor Elevation", 7 "Loader Arrival", and 7 "Loader Stopped In FrontLoading Area"



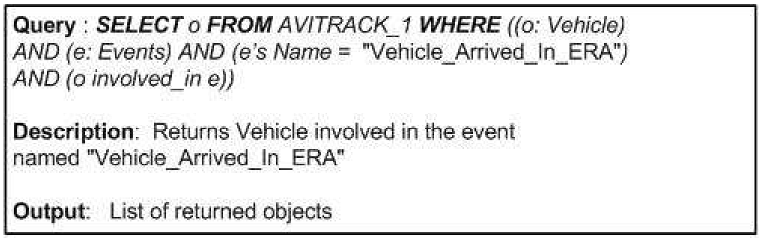
Query : ***SELECT*** *e* ***FROM*** *AVITRACK_1* ***WHERE*** *((o: Vehicle) AND (e: Events) AND (o involved_in e))*

Description: Show events in AVITRACK_1 video concerning Vehicle

Output: List of returned events

Fig. 22. Query 7 returns events in the AVITRACK_1 concerning Vehicle class

events. Query 8 returns 1 Vehicle involved in "Vehicle_Arrived_In_ERA" event.



Query : ***SELECT*** *o* ***FROM*** *AVITRACK_1* ***WHERE*** *((o: Vehicle) AND (e: Events) AND (e's Name =* "Vehicle_Arrived_In_ERA"*) AND (o involved_in e))*

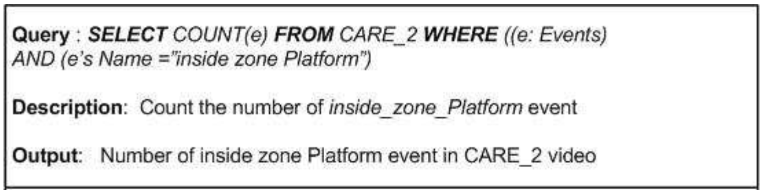
Description: Returns Vehicle involved in the event named "Vehicle_Arrived_In_ERA"

Output: List of returned objects

Fig. 23. Query 8 returns the vehicle involved in the event named "Vehicle_Arrived_In_ERA"

6.3.3. *Queries at the event level*

Queries at the event level can be done based on the attributes of event or the temporal relations between events. Query 9 (Fig. 24) is dedicated for computing statistical information. It enables to count the number of instance of a specified event (event name is set by Inside_Zone_Platform). This query returns 29 events of inside_zone_Platform as described in Tab. 6.



Query : ***SELECT*** *COUNT(e)* ***FROM*** *CARE_2* ***WHERE*** *((e: Events) AND (e's Name ="inside zone Platform"))*

Description: Count the number of *inside_zone_Platform* event

Output: Number of inside zone Platform event in CARE_2 video

Fig. 24. Query 9 counts the number of inside_zone_Platform in the CARE_2 video

6.4. *Discussion*

According to the obtained results this section analyzes good properties and drawbacks of the proposed approach.

6.4.1. *Performance analysis*

In this paper, we have presented both query language usage and retrieval results. Concerning the query language usage, this approach is expressive and expendable. The expressiveness is determined by the number of different queries that users can make while using the language. A number of queries and their combination at different abstraction levels have been introduced. The first experimentation have shown that the system was adapted to the requirements of Roma and Torino subway managers in the framework of CARETAKER project. A visualization interface will be implemented in the near future to be able to evaluate the language with users from other domains.

The effectiveness of a retrieval approach is measured by the relevance of the results. As presented in the previous section, most of the obtained average normalized ranks are small. The obtained average normalized ranks in Fig. 15, Fig.17 by both the proposed approach and the label-based approach show that the proposed approach gives the same results as the label-based search approach in the case that objects are successfully detected and tracked whereas it gives much better results than the label-based approach when these objects are imperfectly detected and tracked. A comparison with the approach of Calderara et al. (see Fig. 7, Fig. 9 and Fig. 10) shows the effectiveness of the proposed approach for working with imprecise indexing.

The proposed approach is based on a video analysis module. Its retrieval performance is closely dependent on the quality of video analysis. An evaluation of the approach over different video analysis algorithms should be done to quantify the dependent impact of vision errors on video surveillance retrieval performance.

6.4.2. *Complexity analysis*

The computation time is an important factor for an indexing and retrieval approach.

In the indexing phase, the computation time of the affine covariant region detection and SIFT extraction algorithms are the most expensive parts. Affine covariant regions include Shape Adapted (SA) and Maximally Stable (MS) regions. According to [20], for the SA region detection algorithm, the complexity of the algorithm computing initial points is $O(n)$ where n is the number of pixels while the complexity of the automatic scale selection and shape adaptation is $O((m+k)p)$ where p is the number of initial points and m is the number of investigated scales in the automatic scale selection and k is a number of iterations in the shape adaptation algorithm. For the MS region detection algorithm, the computational complexity of the sorting step is $O(n)$ and the complexity of the union-find algorithm is $O(nlog(logn))$ while n is the number of pixels. Computing SIFT descriptor includes the orientation assignment and 16 histograms of 8 bins generation. The computation time SIFT for K affine covariant regions can be roughly estimated by $O(K)$. For one video containing N detections of objects (blobs), total computation time in the worst case is $O(N(nlog(logn)))$. In the proposed approach, in spite of using the global image, we detected the affine covariant regions only on the blobs corresponding to physical objects. The size of these blobs is much smaller than the image size. The average blob size is 1322 pixels while the size of image is about 414000 pixels (720×576).

In the retrieval phase, the major computation time is the matching time. It is measured by the time for computing the distance of a pair of blobs and that of two physical objects. For computing the distance between a pair of blobs, by using the approximate nearest neighbors matching the computation time for a query with N indexed objects is $O(N(\ k\ d\ logn))$ where n is the number of detected affine covariant regions, d is the dimensions of space ($d = 128$) and k is the number of nearest neighbors [3]. The time for compute the distance of two physical objects is the time for finding solution of EMD by using the linear programming. Moreover, we reduce the computation time by using the representative blobs (the number is much smaller that that of all blobs).

7. Conclusions and Future work

In this paper, we have presented an approach for surveillance video indexing and retrieval. A flexible and rich query language has been proposed. A data model that allows to work at different abstraction levels has also been introduced. In order to face imprecise and incomplete data indexing, we have introduced an improvement for the representative blob detection algorithm of Ma et al. [18] in the indexing phase and a new object matching based on the EMD in the retrieval phase. The proposed approach has been compared with that of Calderara et al. and has been proved effective for working with the imprecise indexing. The user-centered search in this approach enables users to define their own queries. The obtained results in both query language usage and retrieval show that this approach is able to answer the 5 challenges presented in the introduction.

However, in this approach, there are two main remaining problems. The first problem is the focus of this approach on the object feature similarity. Semantic distance between object classes, events and sub-events should be considered. The second one is the dependency on video analysis module. Therefore, the retrieval performance is closely related to the quality of video analysis module that unfortunately has been proved to be imperfect in some situations [22,23]. We believe that using only results of vision algorithms is not always sufficient to obtain an efficient indexing and retrieval; learning and sharing knowledge with users may improve the retrieval performance. We plan to improve the proposed approach by:

- adding other kinds of approximate matching: in this paper, we have presented an matching algorithm based on object features; This algorithm can

be extended to take into account the semantic distance between concepts in an ontology as proposed by Corby et al. in [8];
- integrating relevance feedback: enabling user interaction also will be considered to improve the retrieval aspect of the proposed approach.

8. Acknowledgments

In this paper, we are using results from the AVITRACK STREP (Specific Targeted Research Project) and the CARETAKER EU IST FP6-027231 project. We would like to thank the people involved in both projects for sharing their results.

References

1. J. F. Allen, "Maintaining knowledge about temporal intervals", *Communications of the ACM* **26(11)** (1983) 823–843.
2. A. Alper Yilmaz, O. Javed and M. Shah, "Object tracking: A survey", *ACM Computing Surveys (CSUR)* **38(4)** (2006) 1–45.
3. S. Arya, D.M. Mount, N. S. Netanyahu, R. Silverman and A. Y. Wu, "An optimal algorithm for approximate nearest neighbor searching fixed dimensions", *Journal of the ACM* **45(6)** (1998) 891–923.
4. A. Avanzi, F. Brémond, C. Tornieri and M. Thonnat, "Design and Assessment of an Intelligent Activity Monitoring Platform", *EURASIP Journal on Applied Signal Processing, special issue in "Advances in Intelligent Vision Systems: Methods and Applications"* (2005) 2359–2374.
5. S. Calderara, R. Cucchiara and A. Prati, "Multimedia Surveillance: Content-based Retrieval with Multicamera People Tracking", in *ACM International Workshop on Video Surveillance & Sensor Networks (VSSN'06)*, Santa Barbara, California, USA, 27 October 2006, pp. 95–100.
6. X. Chen and C. Zhang, "An Interactive Semantic Video Mining and Retrieval Platform–Application in Transportation Surveillance Video for Incident Detection", in *Sixth International Conference on Data Mining*, Dec 2006, pp. 129–138.
7. I. Cohen, Y. Ma and B. Miller, "Associating Moving Objects Across Non-overlapping Cameras: A Query-by-Example Approach", in *IEEE Conference on Technologies for Homeland Security*, Waltham, MA, USA, 2008, pp. 566–571.
8. O. Corby, R. Dieng-Kuntz and C. Faron-Zucker, "Querying the Semantic Web with the CORESE search engine", in *Proc. of the 16th European Conference on Artificial Intelligence (ECAI'2004)*, subconference PAIS'2004, Valencia, 22-27 August 2004, IOS Press, pp. 705–709.
9. N. Durak and A. Yazici, "Multimodal Video Database Modeling, Querying and Browsing", in *International Symposium on Computer and Information Sciences (ISCIS)*, LNCS 3733, 2005, pp. 802–812.
10. N. Ghanem, D. DeMenthon, D. Doermann and L. Davis, "Representation and Recognition of Events in Surveillance Video Using Petri Nets", in*Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04)*, **7**, 2004.
11. A. B. Hampapur, L. Feris, R. Senior, A. C. F Shu, Y. Tian, Y. Zhai and L. Max, "Searching surveillance video", in *IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS 2007)*, 5-7 Sept 2007, pp. 75–80.
12. W. Hu, D. Xie, Z. Fu, W. Zeng and S. Maybank, "Semantic-Based Surveillance Video Retrieval", *IEEE Transactions on Image Processing* **16(4)** (2007) 1168–1181.

13. G. L. Foresti, L. Marcenaro and C. S. Regazzoni, "Automatic detection and indexing of video-event shots for surveillance applications", *IEEE Transactions on Multimedia* **4(4)** (2002) 459–471.
14. F. Fusier, V. Valentin, F. Brémond, M. Thonnat, M. Borg, D. Thirde and J. Ferryman, "Video Understanding for Complex Activity Recognition", *Machine Vision and Applications Journal* **18** (2007) 167–188.
15. M. S. Lew, N. Sebe and R. Jain, "Content-Based Multimedia Information Retrieval: State of the Art and Challenges", *ACM Transactions on Multimedia Computing, Communications and Applications* **2(1)**(2006) 1–19.
16. D. Lowe, "Distinctive image features from scale invariant keypoints", *International Journal Computer Vision* **60(2)** (2004) 91–110.
17. D. G. Lowe, "Object Recognition from Local Scale-Invariant Features", in *Proc. of the International Conference on Computer Vision (ICCV)*, 1999, pp. 1150–1157.
18. Y. Ma, B. Miller and I. Cohen, "Video Sequence Querying Using Clustering of Objects' Appearance Models", in *International Symposium on Visual Computing (ISVC'07)*, November 26-28, 2007, pp. 328–339.
19. J. Meessen, X. Desurmont, J. F. Delaigle, C. De Vleeschouwer and B. Macq, "Progressive Learning for Interactive Surveillance Scenes Retrieval", in *Workshop on Visual Surveillance (VS07)*, 2007, pp. 1–8.
20. K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir and L. V. Gool, "A comparison of affine region detectors", *International Journal Computer Vision* **65(1/2)** (2005) 43–72.
21. H. Müller, S. Marchand-Maillet and T. Pun, "The truth about corel - evaluation in image retrieval", in *In Proc. Of. CIVR*, London, July 2002, pp. 28–49.
22. J. C. Nascimento and J. S. Marques, "Performance evaluation of object detection algorithms for video surveillance", *IEEE Transactions on Multimedia* **8(4)** (2006) 761–774.
23. A.T Nghiem, F. Bremond, M. Thonnat and V. Valentin, "ETISEO, performance evaluation for video surveillance systems", *Proceedings of IEEE International Conference On Advanced Video and Signal Based Surveillance (AVSS)* , London (United Kingdom), September 2007.
24. J. L. Patino, H. Benhadda, E. Corvee, F. Bremond and M. Thonnat, "Extraction of Activity Patterns on large Video Recordings", *IET Computer Vision* **2(2)** (2008) 108–128.
25. K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors", *PAMI* **27(10)** (2005) 1615–1630.
26. M. Petkovic and W. Wonker, "An Overview of Data Models and Query Languages for Content-based Video Retrieval", in *Proceedings of the International Conference on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet*, l'Aquila, Italy, 6 August 2000.
27. P. Remagnino, G. A. Jones, N. Paragios and C. S. Regazzoni, *Video Based Surveillance Systems Computer Vision and Distributed Processing*, Kluwer Academic Publishers, 2002.
28. L. A. Rowe and R. Jain, "ACMSIGMMretreat report on future directions in multimedia research", *ACMTrans. Multimedia Comput. Comm. Appl.* **1(1)** (2005) 3–13.
29. Y. Rubner, C. Tomasi and L. J. Guibas, "A Metric for Distributions with Applications to Image Databases", in *Proceedings of Int. Conf. on Computer Vision (ICCV'98)*, 1998, pp. 59–66.
30. Y. Rui and T. S. Huang, "Relevance feedback techniques in image retrieval", *Principles of Visual Information Retrieval*, Springer-Verlag,(2001) 219–258.

31. J. Stöttinger, N. Sebe, T. Gevers and A. Hanbury, "Colour Interest Points for Image Retrieval", in *Computer Vision Winter Workshop*, St. Lambrecht, Austria, February 6, 2007, pp. 83–91.
32. J. Sivic and A. Zisserman, "Video Google: Efficient Visual Search of Videos," *Toward Category-Level Object Recognition*, LNCS 4170, 2006, pp. 127–144.
33. E. Stringa and C. S. Regazzoni, "Real-time video-shot detection for scene surveillance applications", *IEEE Transactions on Image Processing* **9(1)** (2000) 69–79.
34. M. J. Swain and D. H. Ballard,"Color indexing", *International Journal of Computer Vision* **7(1)** (1991) 11–32.
35. V. T. Vu, F. Brémond and M. Thonnat, "Automatic Video Interpretation: A Novel Algorithm for Temporal Scenario Recognition", in*Proc. of International Joint Conference on Artificial Intelligence (IJCAI'03)*, August 9-15, Acapulco, Mexico, 2003, pp. 1295–1302.
36. D. Xie, W. Hu, T. Tan and J. Peng, "Semantic-based traffic video retrieval using activity pattern analysis", in *International Conference on Image Processing*, **1**, 2004, pp. 693–696.
37. Z. Xiong, X. S. Zhou, Q. Tian, R. Rui and T. S. Huang, "Semantic Retrieval of Video - Review of research on video retrieval in meetings, movies and broadcast news, and sports", *IEEE Processing Magazine* **3(2)** (2006) 18–27.
38. J. S. C. Yuk, K. Y. K. Wong, R. H. Y. Chung, K. P. Chow, F. Y. L. Chin and K.S.H. Tsang, "Object-Based Surveillance Video Retrieval System with Real-Time Indexing Methodology", in *International Conference on Image Analysis and Recognition (ICIAR'07)*, 5-7 Sept 2007, pp. 626-637.