



Video/Audio Networked surveillance system enhAncement through Human-cEntered adaptIve Monitoring

**Large-scale integrating project
Grant Agreement n°248907
01/02/2010 – 31/07/2013**

**Contractual delivery date: January 31, 2011
Actual delivery date: February 7, 2012**

Deliverable D6.2 Report on offline collective behavior building and long-term scene understanding (version 1)

D6.2

**Version: 2.5
Author: INRIA
Contributors: -
Reviewers: MULT - TCF
Dissemination level: PU
Related document(s): -
Number of pages: 29**

Document information

Ver.	Date	Changes	Author (partic.)
1.0	12/08/2011	Creation	J-L. Patino (INRIA)
1.0	27/01/2012	First version	J-L. Patino (INRIA)
1.5	27/01/2012	Inclusion of DB evaluation and properties	H. Falciani (INRIA)
1.5	27/01/2012	Document restructuring	F. Bremond (INRIA)
2.0	06/02/2012	Update DB evaluation	H. Falciani (INRIA)
2.5	06/02/2012	Update long term analysis and final integration	J-L. Patino (INRIA)

Ver.	Date	Approval/Rejection Decision/Comments	Author (partic.)
2.5	06.02.2012	Approval	B. Ravera (TCF)
2.5	06.02.2012	Approval	C. Carincotte (MULT)

Filename convention is defined as follow:

1. **Project number:** VANAHEIM-FP7-248907
2. **Leading participant acronym (MULT, GTT, IDIAP ...):** xxx
3. **Type of document:**
 - Working Document (by default) **WD**
 - Meeting Minutes **MM**
 - Management Report **MR**
 - Activity Report **AR**
 - Deliverable **DR**
4. **Distribution:**
 - Public (PU) **PU**
 - Consortium restricted (CO) **CO**
5. **Serial number (one letter + 2 digits corresponding to the task, deliverables or meeting number):**
 - Deliverables **D**
 - Meeting **M**
 - Report **R**
6. **Revision number:**
 - draft **d**
 - approved **a**
 - version sequence (one digit)

Copyright

© Copyright 2010, 2013 the VANAHEIM Consortium

Consisting of:

Coordinator: Multitel asbl (MULT)	Belgium
Participants: Gruppo Torinese Trasporti (GTT)	Italy
Institut Dalle Molle d'Intelligence Artificielle Perceptive (IDIAP)	Switzerland
Institut National de Recherche en Informatique et en Automatique (INRIA)	France
Rgie Autonome des Transports Parisiens (RATP)	France
Thales Communications (TCF)	France
Thales Italia (THALIT)	Italy
University of Vienna (UNIVIE)	Austria

This document may not be copied, reproduced, or modified in whole or in part for any purpose without written permission from the VANAHEIM Consortium. In addition to such written permission to copy, reproduce, or modify this document in whole or part, an acknowledgement of the authors of the document and all applicable portions of the copyright notice must be clearly referenced.

All rights reserved.

This document may change without notice.

1 Executive Summary

This report describes techniques and results for off-line collective behaviours (T6.2) and on-line learning (incremental learning) of collective behaviours and human activities (T6.3). The first part will describe the techniques used to build collective behaviours by analysing offline large amounts of data. Next we will detail how online learning is employed to perform continuous adaptive scene understanding.

Contents

1	Executive Summary	4
2	Introduction	7
3	General overview of the behaviours building system	8
4	Activity Analysis and Clustering	10
4.1	Trajectory analysis process	10
4.2	Context analysis process	11
4.2.1	Clustering of tracklet entry/exit points	11
4.2.2	Merging tracklet entry/exit zones	12
4.3	Event Analysis process	13
4.4	Statistical Analysis process	13
5	Results	15
6	Evaluation of the proposed approach	20
6.0.1	Video data and metadata	20
6.0.2	Evaluated Algorithms	21
6.0.3	Evaluation processes	21
6.0.4	Evaluation results	23
6.1	Database Processing: Centralized Consistent repository	24
6.1.1	Automated performance evaluation	25
6.1.2	Structuring and building aggregates	26
6.1.3	Spatial operator based processes	26
7	Conclusion	27
8	References	28

List of Figures

1	VANAHEIM general architecture. The Mid/Long-term analysis works on an autonomous way but can also receive queries from the end-user. The module reads its input and stores its output on the Metadata DB.	8
2	Detailed view of the Metadata DB employed for Long-term analysis.	9
3	Detailed view of the Metadata DB employed for Long-term analysis.	10
4	VANAHEIM observed scene: the DiciottoDicembre station hall	15
5	Left panel: Original underground scene observed by the camera with user defined areas delimitating the scene. Right panel: Top view of the observed scene.	16
6	Learned zones correspond to activity areas as discovered with our algorithm. Different granularity levels allow understanding the activity with different resolutions	16
7	Most transitted zones. Color intensity corresponds to number of detected mobiles in the zone.	16
8	Online learning process to maintain learned zones of activity up to date.	18
9	learned zones online at two different days.	19
10	Activity evolution at the vending machine1 for the same period at two different days. . .	19
11	Left panel: Overlapping between learned zones and reference zones. Right panel: Renaming of learnt zones.	22
12	Database processing organisation.	24
13	Automated performance evaluation organisation.	25

List of Tables

1	The three different generated semantic tables.	14
2	Activity reporting (not exhaustive) related to the vending machines area.	17
3	Most common / rare activities.	17

2 Introduction

One of the main scientific challenges in VANAHEIM is the unsupervised clustering and learning of behavioural cues and/or patterns of activities. More precisely, we target to achieve collective behaviour recognition and definition in automatic and domain-independent manner, thus stepping towards a system suitable for analysing large amount of data from a variety of domains.

Because of the complexity of the task, most learning techniques in the domain of behaviour analysis have focused on object recognition [MG06, Lap06, DTS06], and on primitive event recognition [XGM05, LCSL07, SP06, PC03, PFS05a] mainly through statistics computation. For behaviour recognition, three main categories of learning techniques have been investigated.

- The first class of techniques learns the parameters of video understanding programs [HEC06, MF03]. These techniques have been widely used in case of event recognition methods based on neural networks [FMS04], naive Bayesian classifiers [SS05, LSW⁺06] and HMMs [GJH, WB01, ABF06].
- The second class consists in using unsupervised learning techniques to deduce abnormalities from the learnt events [ZGPBM05, FGR03, XG05, XG]. Eigenvector analysis is another unsupervised technique employed to deduce common behaviours, or Eigenbehaviors as they are called [EP09].
- The third class of methods focuses on learning behaviour based on trajectory analysis. This class is the most popular learning approach due to its effectiveness in detecting normal/abnormal behaviours; for instance, on abnormal trajectory detection on roads [PFS05b, SG00] or pedestrian trajectory characterisation [AC07]. Hidden Markov Models (HMM) have also been employed to detect different states of pre-defined normal behaviour [BKS07, Por04].

However, in VANAHEIM, we are interested in extracting meaningful activity clusters from the operational point-of-view, which differs from simple trajectory clusters and normal/abnormal behaviour extraction.

In VANAHEIM, our purpose is to learn all available human-centred features and events, so as to generate clusters that can be relevant from the management and planning point of view. To do so, we are learning not only the main trajectory (i.e. routes) and their characteristics (e.g. speed, proxemic information...) but also the real activities identified (people buying tickets, going through the turnstiles, using escalators, waiting at the platform...). The system thus starts in a first step by the unsupervised learning of the main activity areas of the scene. We employ trajectory-based analysis of mobiles in the video to discover the points of entry and exit of mobiles appearing in the scene and ultimately deduce the different areas of activity. In a second step, mobile objects are then characterised in relation to the learned activity areas. Two kind of behaviours can then be defined either 'staying in a given activity zone' or 'transferring from an activity zone to another' or a sequence of the previous two behaviours. To obtain a robust model of the activity, we are analysing long periods of video recording.

In addition, so as to enable dynamic adaptation to unexpected or newly observed data, we need a system able learn the activity clusters in an on-line way. On-line learning is indeed an important capability required to perform behaviour analysis on long-term basis. For instance, if the clusters have been built with data observing morning activities, evening activities can be miss-analysed and wrongly merged with the already built clusters. To solve this, we propose to control cluster learning rate with coefficients indicating how flexible the cluster can be updated with new data. This propagation is performed locally and incrementally to enable real-time processing.

To evaluate the results of the global system, we will propose new evaluation criteria to better measure the dispersion and overlapping rate between expert-built ground-truth and learned clusters.

The remainder of this document is structured as follows. An architectural overview of the collective behaviours building system is given in section 3. The methodology for activity extraction is presented in section 4. How we evaluate the proposed approach is explained in section 6. Finally, Section 7 draws the main conclusions and describes some perspectives.

3 General overview of the behaviours building system

In VANAHEIM, we are interested in extracting meaningful activity clusters from the operational point-of-view. Our purpose will be thus to learn all available human-centred features and events, so as to generate clusters that can be relevant from the management and planning point of view. To do so, we plan to learn the main trajectories (i.e. routes) and their characteristics (e.g. speed, proxemic information...) but also the real activities identified (people buying tickets, going through the turnstiles, using escalators, waiting at the platform...). For the long-term analysis, large amounts of data corresponding to these features will be stored on different purpose-oriented databases and analysed offline. The figure below depicts the global architecture of VANAHEIM.

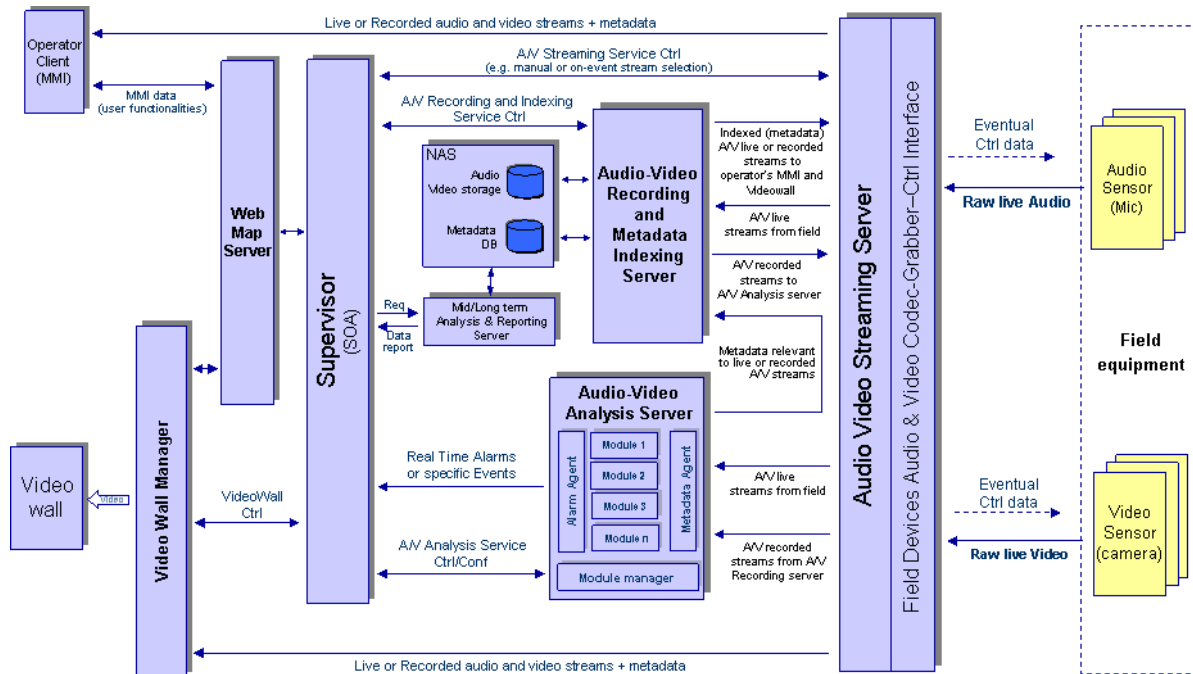


Figure 1: VANAHEIM general architecture. The Mid/Long-term analysis works on an autonomous way but can also receive queries from the end-user. The module reads its input and stores its output on the Metadata DB.

The figure below contains more detailed information on the structure of the Metadata DB. This is actually composed of three specific-purpose databases.

1. Audio-Video Metadata DB: The meta-data contained in this database consists of the data characterizing the persons and events themselves. This information is updated following the real time processing of the audio/video streams.
2. Mid-term Activity DB: The results written to this DB consists of a series of statistics on the number of persons, the trajectory prototypes corresponding to the main flows, and events observed within relatively mid/short analysis periods (a couple of hours) plus.
3. Long-term Activity DB: Analysis of trajectories, events and other features stored in the mid-term DB are employed to find activity patterns not necessarily previously defined. The result will be

written into the long-term DB

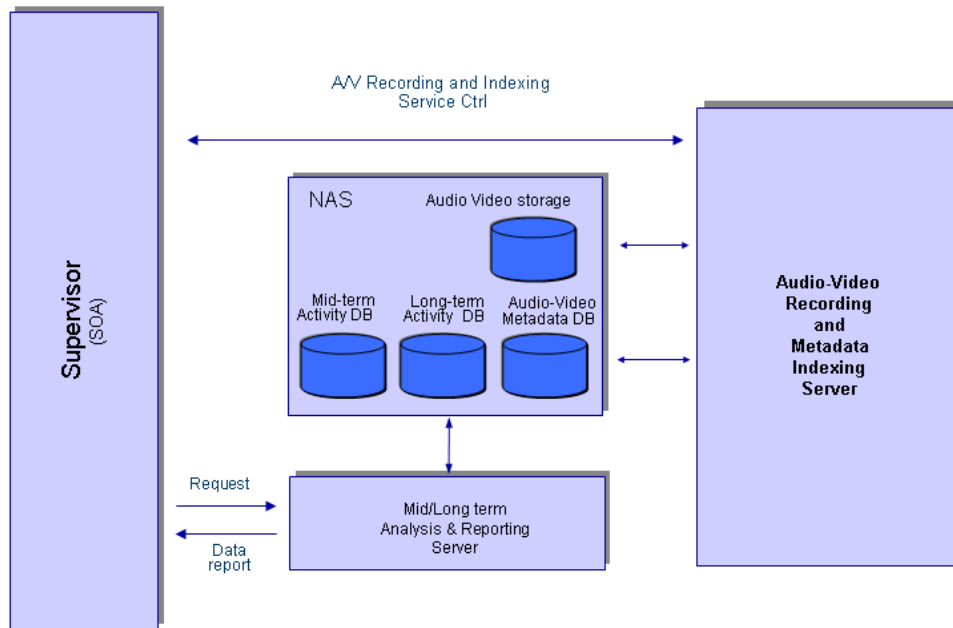


Figure 2: Detailed view of the Metadata DB employed for Long-term analysis.

A more functional view, showing the different analysis processes and interconnections with the different databases, is further shown in the figure below. Real-time components for object tracking and event recognition send their output to a converter module, which places the data into the Metadata DB. From there, there the Long-term analysis module will take its input and initiate the analysis.

The Long-term analysis module will take its input from the metadata DB and sequentially carry out the following processes:

1. **Trajectory analysis process:** we perform here the analysis of trajectories. We study the mobile speed variations in order to distinguish between the mobile in a stationary state or moving state. The former one is more likely to give account of the interactions occurring between mobile objects themselves and interactions between mobiles and the environment.
2. **Event Analysis process:** we aim here to index all events happening in the scene so that in a later stage we can relate them to other features of detected mobile objects. Indexed events can be behaviour events discovered by relating mobile trajectories to learned zones or other human centered events gathered in the VANAHEIM system from other analysis algorithms.
3. **Context analysis process:** We aim here to automatically discover unknown activity zones of the scene and build occupancy statistics having the whole topology at hand.

When these three processes are achieved, the results are written into the mid-term DB. Two subsequent processes are then initiated:

4. **Activity analysis process:** Three inputs are principally taken here. Trajectory type, Event and other human centered features to build clusters of mobile objects involved in a common activity.
5. **Statistical analysis process:** meaningful descriptive measures on the discovered activities and occupied zones are available to the end-user.

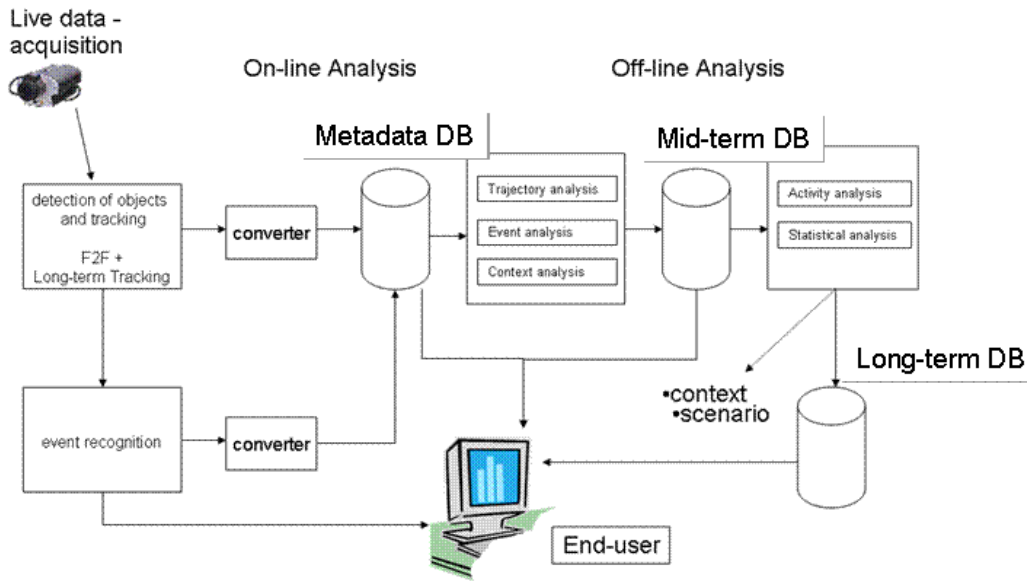


Figure 3: Detailed view of the Metadata DB employed for Long-term analysis.

The whole system helps the manager or designer who wants to get global and long-term information from the monitored site. The user can query the system and specify a period of time where he/she wishes to retrieve and analyse stored information. Moreover, the discovered information on activity should help to update scenario information required for event recognition. In this document we report the current work, which has been addressed for processes 1,2,3 and 5. Process 4 is the subject of our future work.

4 Activity Analysis and Clustering

We understand activity as the interactions occurring between mobile objects themselves and those between mobiles and the environment. We employ trajectory-based analysis of mobiles in the video to discover the points of entry and exit of mobiles appearing in the scene and ultimately deduce the different areas of activity. In a second step, mobile objects are then characterised in relation to the learned activity areas.

4.1 Trajectory analysis process

In order to discover meaningful activity, it is of prime importance to have available detailed information allowing to detect the different possible interactions between mobiles. As our system is based on trajectory analysis, the first step to prepare the data for the activity clustering methodology is to extract tracklets of fairly constant speed allowing to characterise the displacements of the mobile or its stationary state.

If the dataset is made up of N objects, the trajectory tr_j for object O_j in this dataset is defined as the set of points $[x_j(t), y_j(t)]$ corresponding to their position points; x and y are time series vectors whose length is not equal for all objects as the time they spend in the scene is variable. The instantaneous speed for that mobile at point $[x_j(t), y_j(t)]$ is then $v(t) = (\dot{x}(t)^2 + \dot{y}(t)^2)^{\frac{1}{2}}$. The objective is then to detect those points of changing speed allowing to segment the trajectory into tracklets of fairly constant speed

so that the trajectory can be summarised as a series of displacements at constant speed or in stationary state.

The mobile object time series speed vector is analysed in the frame of a multiresolution analysis of a time series function $v(k)$ with a smoothing function, $\rho_{2^s}(k) = \rho(2^s k)$, to be dilated at different scales s . In this frame, the approximation A of $v(k)$ by ρ is such that $A_{2^{s-1}}v$ is a broader approximation of $A_{2^s}v$. By analyzing the time series v at coarse resolutions, it is possible to smooth out small details and select those points associated with important changes.

The speed change points are then employed to segment the original trajectory tr_j into a series of i tracklets tk . Each tracklet is defined by two key points, these are the beginning and the end of the tracklet, $[x_j^i(I), y_j^i(I)]$ and $[x_j^i(end), y_j^i(end)]$. By globally reindexing all tracklets, let m be the number of total tracklets extracted, we obtain the following tracklet feature vector :

$$tk_m = [x_m(I), y_m(I), x_m(end), y_m(end)] \quad (1)$$

4.2 Context analysis process

Modeling the spatial context of the scene is essential for recognition and interpretation of activity. Because it is not possible to define a-priori all activity zones, manually defined Contextual zones do not suffice to describe all possible situations or evolving actions in the monitored scene. We thus propose to learn the complementary activity zones.

The feature vector defined in equation 1 constitute a set of simple descriptors that have proven experimentally to be enough to describe activities in a large variety of domains (such as traffic monitoring, subway control, monitoring smart environments), mainly because they define where the object is coming from and where it is going to and also with approximative constant speed.

In our system, mobiles starting (or ending up) their displacement at nearby positions are seen to share a common activity area. Finding thus the activity areas on the scene could be seen as an equivalent task of clustering entry/exit tracklet points. Different common clustering algorithms such as k-means, Self Organising Maps, etc... can be used but the drawback is, first, the number of clusters must be known in advance, and second, the spatial morphology of the resulting clusters is always circular, which may not correspond to the real scene topology. In our approach, we propose to find first high resolved (small in size) entry/exit activity zones, which in a second step could be further regrouped to obtain broader activity zones (following an hierarchical agglomerative algorithm).

4.2.1 Clustering of tracklet entry/exit points

For this first step, we employ the well-known clustering Leader algorithm [Har75]. It has the advantage to work on-line without needing to specify the number of clusters in advance. In this method, it is assumed that a threshold T is given. The algorithm constructs a partition of the input space (defining a set of clusters) and a leading representative for each cluster, so that every object in a cluster is within a distance T of the leading object. The threshold T is thus a measure of the diameter of each cluster. The algorithm makes one pass through the dataset, assigning each object to the cluster whose leader is the closest and making a new cluster, and a new leader, for objects that are not close enough to any existing leaders. Defining T is application dependent. In this work we have employed the threshold setting suggested by [Pat11].

Let's consider the position of a mobile is $L(x,y)$, its influential zone, Z_n , is defined by a radial basis function (RBF) centered at the position L ; and the belongingness of a new point $p(x,y)$ to that zone is given by:

$$Z_n(L, p) = \phi(L, p) = \exp(-\|p - L\|^2 T^2) \quad (2)$$

The RBF function has a maximum of 1 when its input is $p = L$ and thus acts as a similarity detector with decreasing values outputted whenever p strides away from L . An object element will be included

into a cluster Z_n if $Z_n(L, p) \geq 0.5$, which is a natural choice. The cluster receptive field (hyper-sphere) is controlled by the learnt parameter T

4.2.2 Merging tracklet entry/exit zones

We find the final activity areas by aggregating similar entry/exit tracklet zones. We look to establish a similarity relation between the different zones defined by the tracklets. On the end, new zones are given by the fulfillment of two relations: 1) zone Z_{n_i} overlaps zone Z_{n_j} , and 2) zone Z_{n_j} does not overlap any a-priori user-defined contextual Zone Z_{ctx_q} . These relations are defined:

$R1_{ij}$: Zone Z_{n_i} overlaps Zone Z_{n_j}

$$R1_{ij} = \sum_{k=1}^3 \left[\sum_{(x,y) \in (X_{ik}, Y_{ik})} Z_{n_j}(x, y) \right] \quad (3)$$

and $X_{ik} = \left\{ \frac{(k+1)}{3}T \cos(\theta) + L_i(1) \right\}$,

$Y_{ik} = \left\{ \frac{(k+1)}{3}T \sin(\theta) + L_i(2) \right\}$ with $\theta = 0, \dots, \frac{\pi}{8}, \dots, 2\pi$

That is, points $(x, y) \in (X_{ik}, Y_{ik})$ belonging to concentric circles to L_i are tested to verify how they fit Z_{n_j} in order to calculate the overlap/similarity between Z_{n_i} and Z_{n_j} . This allows avoiding equity problems with clusters defined on sparse regions (some clusters may be defined with a much larger number of points than others).

$R2_{iq}$: Zone Z_{n_i} overlaps Zone Z_{ctx_q}

$$R2_{iq} = \sum_{k=1}^3 \left[\sum_{(x,y) \in (X_{ik}, Y_{ik})} Z_{ctx_q}(x, y) \right] \quad (4)$$

It is possible to transform R2 into a new relation, R3, which links Z_{n_i} and Z_{n_j} if both clusters are related to the same Zone Z_{ctx_q} through the fulfillment of R2. The relation between Z_{n_i} and Z_{n_j} is then given by

$$R3_{ij} = \max_q \min [R2_{iq}, R2_{qj}] \quad (5)$$

Remark that $\overline{R3}$, the complement to R3 given by $\overline{R3} = -R3$, represents the relation linking Z_{n_i} and Z_{n_j} if both clusters are not related to any contextual Zone (Z_{ctx_q}). R1 and $\overline{R3}$ can be aggregated employing a soft computing aggregation operator such as $R = R1 \cap \overline{R3} = \max(0, R1 + \overline{R3} - 1)$ and made transitive with:

$$R \circ R(x, y) = \max_z \min [R(x, z), R(z, y)] \quad (6)$$

R is now a transitive similarity relation with R indicating the strength of the similarity. If we define a discrimination level α in the closed interval $[0,1]$, an α -cut can be defined such that

$$R^\alpha(x, y) = 1 \Leftrightarrow R(x, y) \geq \alpha \quad (7)$$

It is thus implicit that $\alpha_1 > \alpha_2 \Leftrightarrow R^{\alpha_1} \subset R^{\alpha_2}$; thus, the R^α form a nested sequence of equivalence relations, or from the classification point of view, R^α induces a partition $\pi^\alpha = \{Z_i^\alpha\}$ of $X \times Y$ (or X^2) such that $\alpha_1 > \alpha_2$ implies π^{α_1} is a refinement of π^{α_2} .

At this point, the difficulty comes down to select the appropriate α -cut such that π^α from R^α represents the best partition of the data. This is still a difficult and open issue that we choose to approach by selecting the alpha-values, which induce a significant change from π^{α_k} to $\pi^{\alpha_{k+1}}$.

To automatically detect those significant partition changes we choose to study the cluster area and number of clusters induced at each partition π^α . We achieve this in the frame of a multiresolution analysis. By analysing induced partitions at coarse resolutions, it is possible to smooth out small details and select the $\alpha - cut$ levels associated with important changes. From the monitored scene, it would be useful to distinguish among different information levels: (i) grouped activity on large spaces, (ii) very detailed individual activity, (iii) somewhere meaningful in-between the last two. For this reason, when performing activity zone discovery, we automatically select the three highest change-inducers $\alpha - cut$ levels from the previous analysis. The result is then that we end up with a three levels hierarchy of activity zones.

4.3 Event Analysis process

We aim at creating a system for the recognition and interpretation of human activity and behaviour, and extract new information of interest for end-users. Low-level tracking information is thus expected to be transformed into high-level semantic descriptions conveying useful and novel information. In our application, we establish a semantic meaning from the scene model built as described in the previous section (4.2). The behaviour knowledge can be thus expressed with semantic concepts, instead of using quantitative data, thanks to the learned zones and any user-defined contextual zone. Let us assume, we have in total $k = 1, \dots, K$ contextual zones on the scene defined either a-priori, or after the zone learning procedure. Two different kinds of behaviour events can then be identified:

- Mobile moving from Zone $Zctx_k$ to Zone $Zctx_{k'}$
- Mobile Inside Zone $Zctx_k$

4.4 Statistical Analysis process

Statistical information can be obtained from the mobile objects and the contextual objects as well as their interactions. This is a major information source for the end-user. For instance, on large metro video recordings, there is spatial and temporal information on the use of contextual objects.

In order to have a clear and compact representation of the human activity evolving on the video, we have divided all related information to objects and events detected on the video into three different semantic tables: mobile objects table, events table and contextual objects table. Some structured knowledge representation had been introduced before but in this contribution we propose a semantic representation which takes also into account interactions between tracked objects in the video and their environment.

Each column in Table 1, presented below, contains the fields that we have included for each semantic table. Statistical information obtained from the semantic tables we have set up allow for a comparative analysis of contextual objects in the scene. The user can then choose the activity area to analyse from all automatically learned zones. The temporal evolution of an activity zone can also be followed as well to add for more descriptive statistics for the end user.

Table 1: The three different generated semantic tables.

Mobile Objects Table	Events Table	Contextual Objects Table
<ul style="list-style-type: none"> - id. The identifier label for the object. - type. The class the object belongs to: Person, Group, Crowd or Luggage. - start. Time the object is first seen. - end. Time the object is last seen. - shape. The label describing the object's shape depending on the object's ratio height/width. - involved_events_id. All occurring Events related to the identified object. - significant_event. The most significant event among all events. This is calculated as the most frequent event related to the mobile object. - trajectory_type. The trajectory pattern characterising the object. 	<ul style="list-style-type: none"> - id. The identifier label for the detected Event. - type. The class where the Event belongs to (i.e. 'close_to', 'stays_at', ...) - start. First moment on which the Event is detected. - end. Last moment on which the Event is seen. - involved_mobile_object_id. The identifier label of the object involved in that event. - involved_ctx_object_id. The name of the contextual object involved in that event. 	<ul style="list-style-type: none"> - id. The identifier label for the object - type. The class of the object - significant_event. The most significant event among all events but referring to contextual objects. - start; - end. refer to the first and last instant the mobile object interacts with the contextual object - involved_events_id. All occurring Events related to the identified contextual object. - rare_event. This is the rarest event. - event_histogram. Gives the frequency of occurrence of all involved events. - involved_mobile_objects_id. All detected mobile objects interacting with the contextual object of interest. - histogram_mobile_objects. Gives the frequency of appearance for all involved mobile objects. - use_duration. Percentage of occupancy (or use of a contextual object). For instance, the Ticket Machine has a 10% of use over the observation time. - mean_time_of_use. Average time of interactions between the mobile object and the contextual object.

5 Results

The algorithm for unsupervised learning of activity areas (as described in section 4.2) was applied to a one hour-duration video recorded at one entrance hall from the Torino underground system. The final relation R given in equation (6), which verifies the transitive closure, is thresholded for different $\alpha - cut$ values going from 0 to 0.9 and with a step value of 0.05. The algorithm automatically selects the best $\alpha - cut$ value giving a detailed composition of the activity areas of the scene as mentioned in section 4.2.2. Additionally, the system gives also the possibility to the end user to select the partition containing the number of activity areas which may suit best his needs. Figure 4 presents the observed scene at the Turin metro station and from the camera we are currently analysing. Figure 5 presents some manually defined zones and a top view representation of the scene. Those user defined areas are:

- zoneMap1 (ZM1)
- zoneMap2 (ZM2)
- zoneVendingMachine1 (ZVM1)
- zoneVendingMachine2 (ZVM2)
- zoneEntrance1 (ZE1) (north entrance)
- zoneEntrance2 (ZE2) (south entrance)
- zoneHall (ZH)
- zoneTurnstiles (ZT)

Finally, figure 7 presents those learned zones corresponding to the analysed video. Four different activity levels are calculated from the system. The first level groups all the activity in the scene and thus gives the information as a single activity area where individuals are moving around in the station. The second level gives the information of activity occurring in broad areas. The four level gives activity information with smaller and more detailed areas. The third level is a compromise between levels two and four. It can be observed that at the different granularities, those areas which are the most employed are: The entry/exit areas to the station, the turnstiles area, and at a lower degree, the areas corresponding to the vending machines. This can be better observed in figure 7 depicting only those activity areas most transitted.



Figure 4: VANAHEIM observed scene: the DiciottoDicembre station hall

As mentioned in section 4.3, we achieve behaviour characterisation by linking low-level tracking to the learned zones. The whole activity observed from the scene can then be reported following the behaviours inferred from the learned zones. For instance, for the ticket machine area (zone Zn5), the obtained report is given in table 2. Zone Zn5 corresponds to the vending machine most used from the

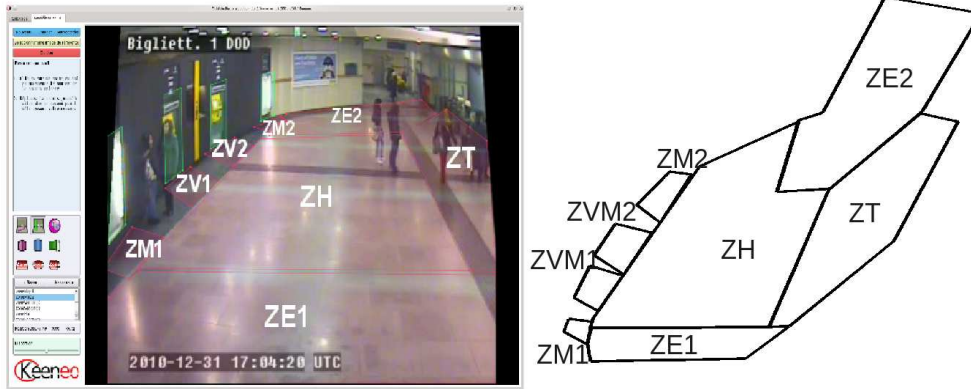


Figure 5: Left panel: Original underground scene observed by the camera with user defined areas delimitating the scene. Right panel: Top view of the observed scene.

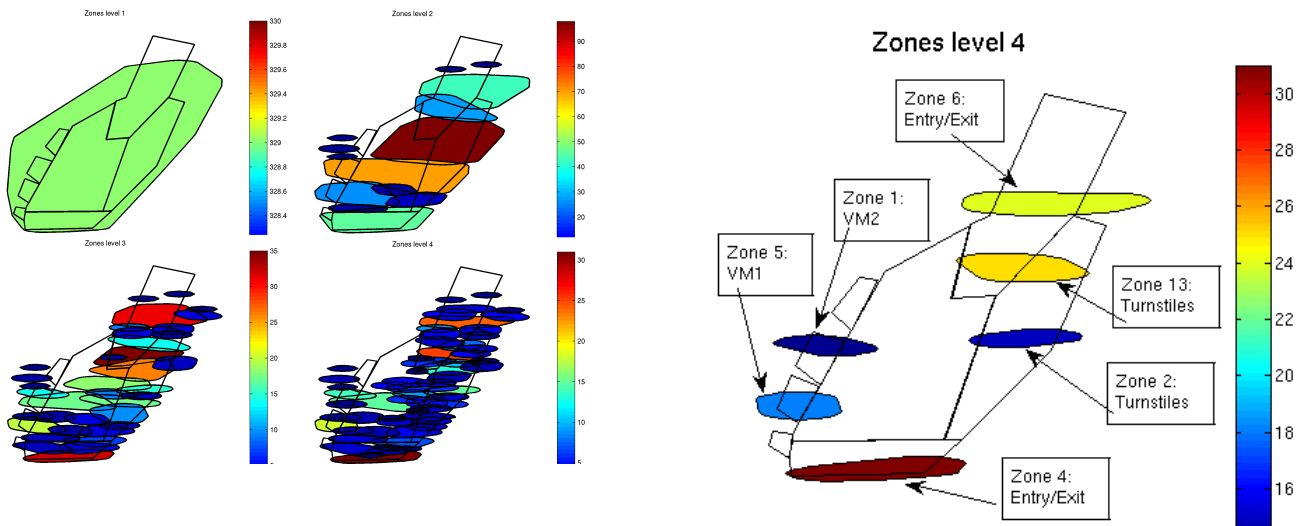


Figure 6: Learned zones correspond to activity areas as discovered with our algorithm. Different granularity levels allow understanding the activity with different resolutions

Figure 7: Most transitted zones. Color intensity corresponds to number of detected mobiles in the zone.

two in the station hall. Most people (7 out of 11) going to this vending machine depart from the Zone Zn4 (the south entry/exit of the station); and people at the vending machine go to the turnstiles to enter the station. While at the vending machine, people tend to spend in the mean 12 seconds.

Table 2: Activity reporting (not exhaustive) related to the vending machines area.

Number of mobiles	Description	Semantic
9	Zone 4 to Zone 5	South entry to VM1
7	at Zone 5	Stays at VM1
4	Zone 5 to Zone 47	VM1 to Turnstiles
1	Zone 5 to Zone 2	VM1 to Turnstiles

When arranging the occurring events according to their semantic, and looking at their frequency of occurrence, it is possible to observe (from table 3) that the most common activities carried out in the station hall are people detected at the north station entry/exit, people detected at the turnstiles area and the flow of people between these two areas (representing already 50% of the whole station activity). On the other hand, rare activities are for instance those of people going through the station from one entry/exit to the opposite entry/exit or going from the Turnstiles area to the vending machines. We have visually (i.e. observing the video) verified the correctness of frequent and rare activities.

Table 3: Most common / rare activities.

Proportion	Number of mobiles	Semantic
29.7%	98	at zone Turnstiles
13.6%	45	at zone South Entry
7.2%	24	zone South Entry to zone Turnstiles
6.0%	20	zone Turnstiles to zone South Entry
5.1%	17	at zone North Entry
...		
0.30%	1	zone Vending machine1 to zone Vending machine2
0.30%	1	zone Turnstiles to zone Vending machine1
0.30%	1	zone North Entry to zone South Entry

In order to cope with activity changes that normally occur when observing long periods of video, activity clusters have to be updated online when new data is available. Because in our approach we achieve behaviour characterization by linking low-level tracking to the learned zones. Keeping our behaviour patterns up to date implies maintaining the set of learned zones also updated with the arrival of new data by aggregating new activity zones if such zones appear and enlarging or diminishing already learnt zones according with the use of the station. This process is graphically explained in figure 8

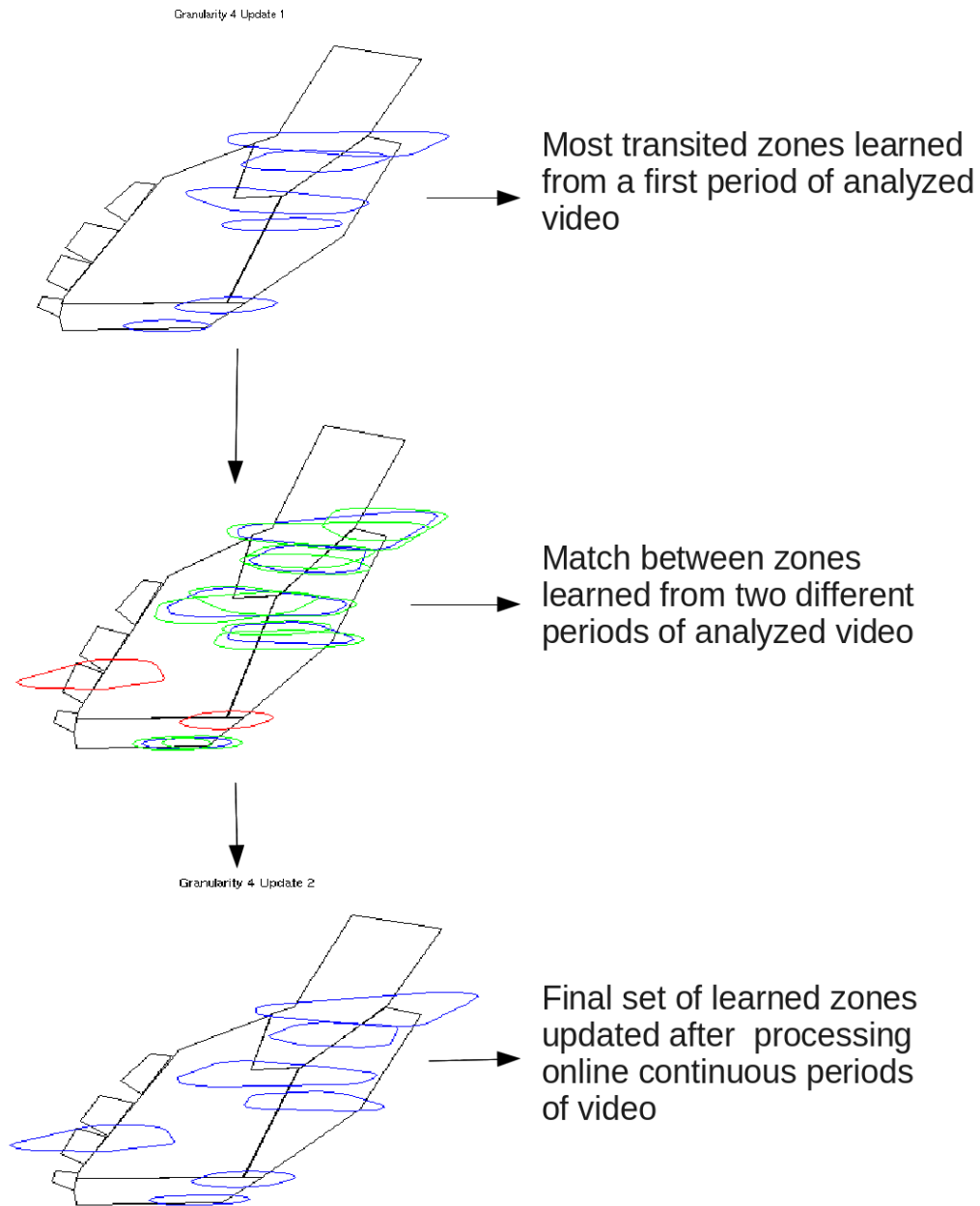


Figure 8: Online learning process to maintain learned zones of activity up to date.

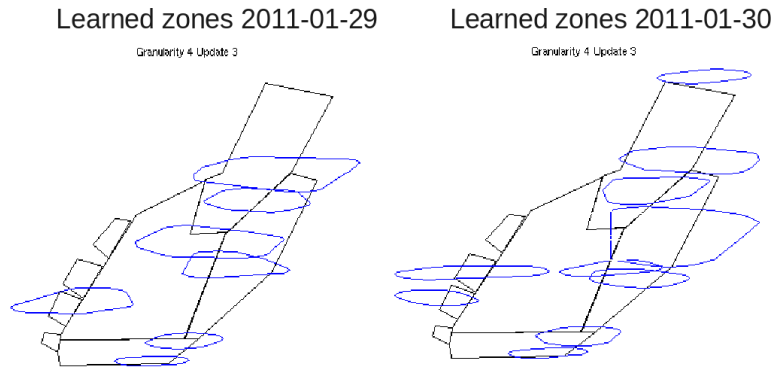


Figure 9: learned zones online at two different days.

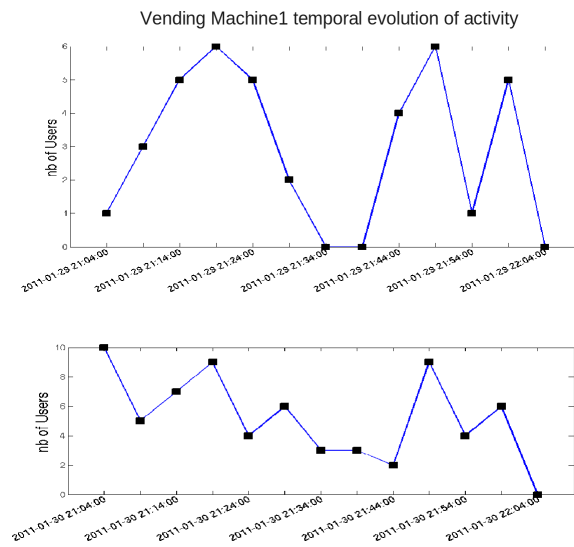


Figure 10: Activity evolution at the vending machine1 for the same period at two different days.

Online learning allow us then to process large amounts of video and calculate statistics which are meaningful for the end user. For instance, learned zones corresponding to entry/exit areas in the station or those corresponding the vending machines are the most interesting to follow in their temporal activity evolution.

Figure 9 shows for instance those activity zones learnt online at two different days and which show some occupancy in the period 21:00 to 22:00 in the evening. As it can be observed, some zones are incrementally added like that corresponding to the vending machines2, which appears not to be occupied for the same period in the first analysed day. Accordingly other zones have shrunk or enlarged given the frequency of travelers in the station at that period.

We can compare a specific activity zone like that corresponding to the vending machine1 in two different days. Figure 10 shows the number of users detected in that activity zone for two consecutive days and at the same period. It can be observed that in the second day the use of the vending machine has clearly increased. This kind of statistics on long time records are of prime importance to the end-user.

6 Evaluation of the proposed approach

The primary objective of this evaluation process is to establish how accurately

- Events are recognized from trajectory endpoints
- Activities are discovered

The evaluation is made directly inside the DB. Rather than manipulating simple database types like string or integer, spatial feature offers in a consistent way points, trajectory and polygons types. After detailing the experiment in terms of video and resources (metadata DB), the evaluation process is exposed.

6.0.1 Video data and metadata

Data set:

1. The evaluation is conducted using 1-hour long video from VAHAHEIN dataset.

Metadata DB:

From these videos, metadata, containing people trajectories, are produced based on the IDIAP people detection and people tracking algorithms.

In this part of the document we refer to this metadata as “metadata DB”.

Zone and activity reference data:

The recognized events are the people motions from one zone to another zone or staying in one zone as stated in 4.3. So we have people

- Entering a zone
- Leaving a zone
- Staying for a while in a zone

The discovered activities are the chain of events like entering, staying and leaving zone for the same person ordered by the time in which the events happen.

The reference data correspond then to eight predefined zones, which are those in figure 5

Based on these zones, reference activity classes are built up from the metadata DB trajectories (called reference trajectories) processed from the one-hour video. One trajectory corresponds to one reference activity class. 213 DB trajectories have been validated visually by human operator. The wrong trajectories have been removed.

The reference activity is an ordered set of the reference zones intersected by a reference trajectory:

An activity A_k is defined if

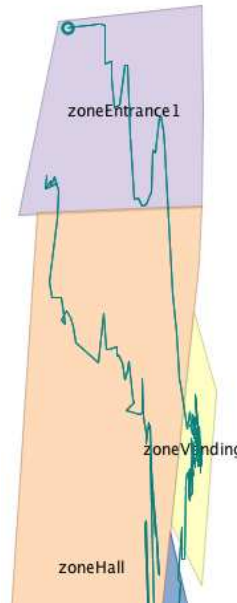
$\exists Traj_k \in \Gamma$, with Γ the set of trajectories

and $Z_{ref_k} \in \Omega_{ref}$ the set of reference zones, such that

$name_{A_k} = \{name(Z_{ref_i}), i \in card(\Omega_{ref}) / Traj_k \cap Z_{ref_i} \neq \emptyset\}$

In the opposite figure we see the trajectory of one person entering in zoneEntrance1 (ZE1), going across zoneVendingMachine1 (ZVM1) and zoneHall (ZH).

So the activity is defined as zoneEntrance1_to_zoneHall_to_zoneVending... zoneEntrance1 label.



6.0.2 Evaluated Algorithms

The first step of the zone discovery algorithm is a clustering algorithm using the leader algorithm.

Different initiation techniques are tested with the Leader Clustering algorithm and are described hereafter:

Iterative initiation technique: As soon as the arriving points are too far, new clusters are created, iteratively.

Grid-based initiation technique: the first leaders are the vertices of the grid covering the scene. The number of vertices is the only parameter to be set (dynamically or not).

Density-based initiation technique: the first leaders are the vertices with the most surrounding neighbors.

There are two implementations of the zone discovery algorithm. The first implementation is done in matlab and the other one in PL/SQL (relational database style implementation). The PL/SQL implementations are the one implementing all different initiation techniques. The matlab implementation contains the first two. In this work we have focused on the iterative initiation technique because it delivers more accurate shaped zones.

In figure 2 a “zoneEntrance1” zone is one of the reference zones and in this case, zones 9,19 and 1 are mapped to “zoneEntrance1”. As a result in figure 2b we see the new names for several learnt zones.

6.0.3 Evaluation processes

The reference zones are built up from reference activities; the output of a discovered activity is one or two zones corresponding to the motion of the people within the observed scene. The evaluation process consists thus in calculating the distance between the discovered zones and the reference zones.

Zone overlapping distance

We define the distance between the reference zones Ω_{ref} and a discovered zone Z_k the overlap between these two zones.

For every discovered zone we search for the reference zone that overlaps the most in term of common area.

$$\forall Z_k \in \Omega, \text{ with } \Omega \text{ the set of learnt zones, at most}$$

$$\exists Z_{ref_k} \in \Omega_{ref}, \text{ with } \Omega_{ref} \text{ the set of reference zones,}$$

such that

$$Area(Z_{ref_k} \cap Z_k) = \max_{i \in Card(\Omega_{ref})} Area(Z_{ref_k} \cap Z_k)$$

Therefore Z_k is renamed with the intersection name and

$$dist(Z_k, Z_{ref_k}) = Area(Z_{ref_k} \cap Z_k) / \max(Area(Z_{ref_k}), Area(Z_k)).$$

In figure 11 a “zoneEntrance1” zone is one of the reference zones and in this case, zones 9,19 and 1 are mapped to “zoneEntrance1”. As a result we see, in the right panel of the figure, the new names for several learnt zones.

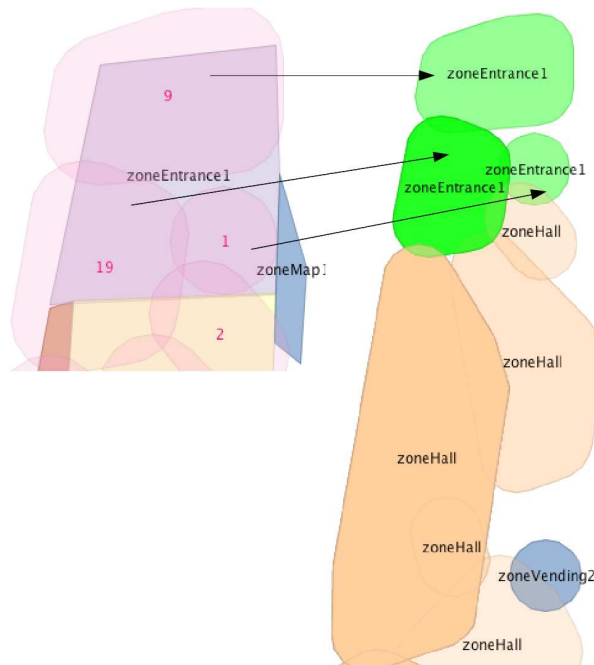


Figure 11: Left panel: Overlapping between learned zones and reference zones. Right panel: Renaming of learnt zones.

We call a True Positive zone (TP) any Z_k such that $dist(Z_k, Z_{ref_k}) > 20\%$

We call a False Positive zone (FP) any Z_k which does not intersect any reference zone.

This FP zone could correspond to an area where rare event occurs and which the user has not modeled.

We call False Negative zone (FN) any reference zone that has no intersection with the discovered zones.

6.0.4 Evaluation results

One-hour sequence	Matlab implementation	Database implementation
Num. True Positives	26	7
Num.False Positives	3	0
Num. False Negatives	1	4
Precision (global)	0.89	1
Sensitivity (global)	0.96	0.63

Precision (P) = TP/(TP+FP)
 Sensitivity (S) = TP/(TP+FN)

Runs results Overall Remarks

- Event detection accuracy is not part of the current evaluation
- Noise reduction (ie trajectory filtering based on number of frames for example) is reported to future works for the sql implementation (done already in matlab)
- The evaluation is highly sensitive to reference zones

Visual Analysis

Both False Positive and False negative are identifiable:

- False Positive visualization:

In the figure Eval01 the trajectory going from zone 16 to zone 26 intersects zone 3. But neither zone 16 nor zone 3 overlap with any reference zone.

- False Negative visualization:

In figure Eval02 the reference zone zoneMap2 matches with no discovered zone enough. The reference zone 26 is overlapping the most with reference zone zoneVending2 instead.

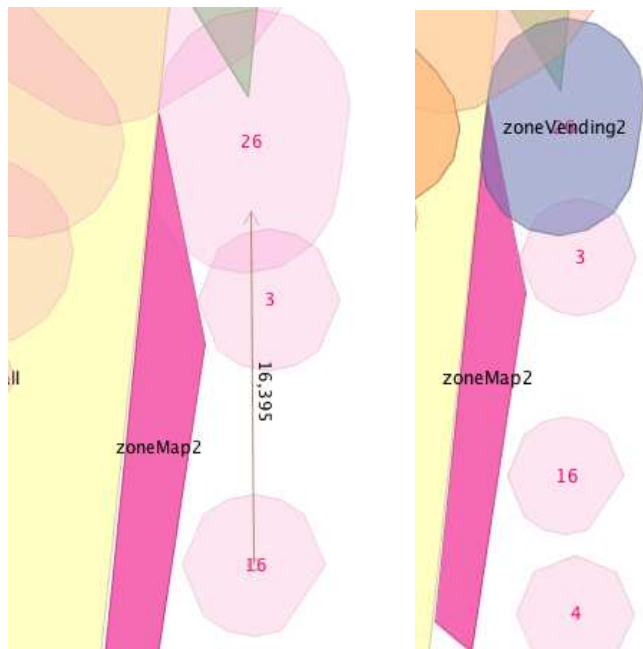


figure Eval01

figure Eval02

6.1 Database Processing: Centralized Consistent repository

The database structuring mechanisms as schema, tables and views offers simple and efficient way To organize data, keeping track of the whole life cycle of formatting, combining and aggregating tasks required by the performance evaluation process.

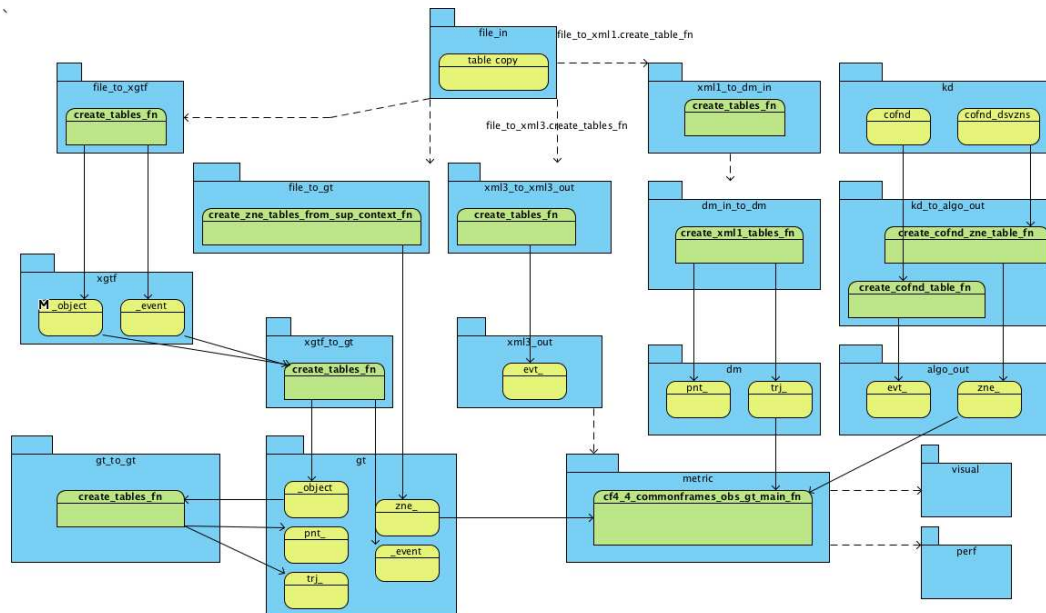


Figure 12: Database processing organisation.

This figure presents the main schema needed to calculate and enhance performance evaluation. Hereafter short details are provided for each of them.

- **file_in**:file_in stands for file inputs. This schema contains all the files, xml or text ones, inserted into database to be processed.
- **file_to_xgft**:This schema contains the functions to select and process xgft files.
- **xml1_to_dm_in**:xml1_to_dm_in stands for "xml1 files to data-mining inputs".
- **kd**:KD schema stands for "Knowledge discovery". It contains all the data produces by the clustering algorithm runs.
- **file_to_gt**: Schema containing all formatting functions required to format ground truth files to be processed by metrics.
- **xml3_to_xml3_out**: Xml3_out is the data repository structures for formatted xml3 data
- **dm_in_to_dm**: dm_in_to_dm is used to produce points and trajectories data structures.
- **kd_to_algo_out**: kd_to_algo_out schema stands for "Knowledge discovery to algorithms output". It contains all the functions implementing formatting of data resulting from algorithms runs. It also contains the logging tables to track eventual errors.
- **xgft**: Contains the data extracted from xgft files. The information are splitted in two tables per file
- **xgft_to_gt**: this schema contains the function to format and export to gt schema the data from xgft files.

- **xml3_out**: this schema contains the formatted data coming from xml3 files.
- **dm**: dm stands for data-mining. This schema contains all the inputs the algorithms will process.
- **algo_out**: Stands for "algorithms output". This schema contains the data produced by algorithm runs.
- **gt_to_gt**: This schema contains the functions to format and enhance the basic data structures with spatial data types
- **gt**: gt stands for "ground truth". It contains the data required to evaluated algorithms.
- **visual**: This schema contains the tables dedicated to be visualized through GIS spatial client
- **metric**: metric schema contains the functions implementing specific metrics.

6.1.1 Automated performance evaluation

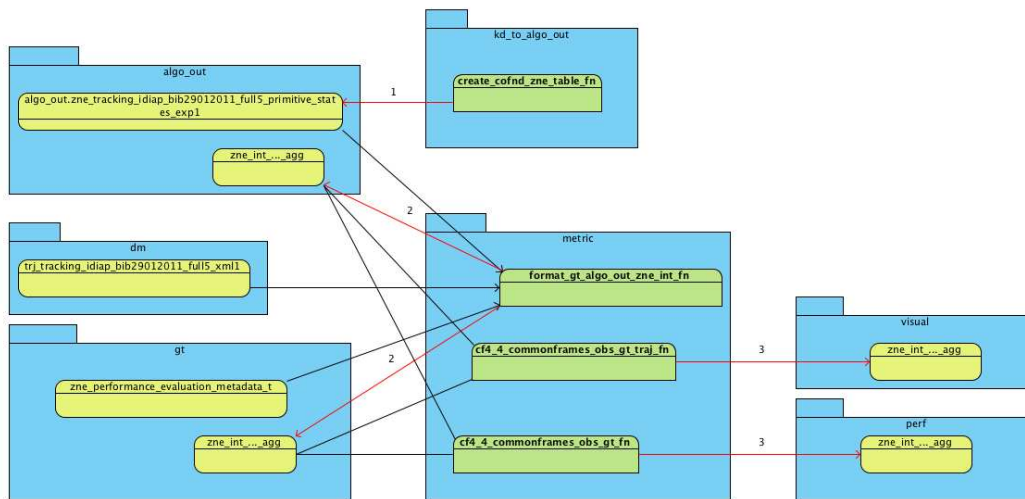


Figure 13: Automated performance evaluation organisation.

- **create_cofnd_zne_table_fn**: this function extract from kd schema the zones defined in the cofnd_dsvzns table. As input parameter is chosen an identification name for the algorithm run.
- **format_gt_algo_out_zne_int_fn**: this function do the matching between the ground truth zones and the learned ones.It results in many discovered zones with the same ground truth zone name.
- **cf4_4_commonframes_obs_gt_traj_fn** this function produce the trajectories and annotate them in term of TP,FP,FN
- **cf4_4_commonframes_obs_gt_fn** this function generate the 4.4 metric results ventilated by event-name as well as consolidated
- **zne_int_..._agg**: zne stands for zone int stands for intersection agg stands for aggregate

6.1.2 Structuring and building aggregates

In addition to traditional metadata belonging to frames, GIS points and GIS lines are used to store coordinates as well as temporal information. The trajectory being a GIS line made of all the points of the trajectory. The M GIS dimension stores the corresponding frameid information.

Then instead of manipulating 15 000 points, corresponding about one hour frames, we end up with a database table with a bit more than 200 lines, one line per mobileid. Furthermore, the zones, both from ground truth and learned ones, are associated with convex hull GIS polygon. Last structure filling the gap with our needs.

6.1.3 Spatial operator based processes

With lines and polygons we can determine overlapping area, intersecting zones and trajectories now. Intersect and overlap GIS spatial operator offer an efficient way to query all the information we have at once in a single pass, with respect to relational closure.

Then in a scalable and easy way it has been possible to enhance matching procedure of concordant events present in ground truth and by the clustering algorithm. Initially, the matching between learned event and ground truth was done truth selecting among learnt zones, from the algorithm, the single one overlapping the most the ground truth zone. Instead, a simple SQL query enforcing a one-to-many relationship sufficed to associate one ground truth zone to many learnt ones.

The SQL expressions extended with spatial operators proved to be enough to implement and tune the whole performance evaluation process in a smooth and self-descriptive way.

7 Conclusion

In this report we have presented the current progress for off-line collective behaviours (T6.2) and on-line learning (incremental learning) of collective behaviours and human activities (T6.3). So far we have setup a system which starts in a first step by the unsupervised learning of the main activity areas of the scene. In a second step, mobile objects are then characterised in relation to the learned activity areas: either as 'staying in a given activity zone' or 'transferring from an activity zone to another' or a sequence of the previous two behaviours if the tracking persists long enough. This characterisation has allowed us to obtain already informative statistics on the use of the station following the employed camera, and discover what are the main activity areas and main displacements of people in the station. Applying on-line learning the system is able to continuously process long-term video recordings. In this document we have shown examples from the analysis of two consecutive days of how activity zones can be learnt incrementally and refined according to the station use. Our current evaluation signals encouraging results and we are progressing towards an efficient sql implementation. Our future work consists in a more complex characterisation of the mobiles by including other features such as timestamp and speed.

8 References

- [ABF06] E L Andrade, S Blunsden, and R B Fisher. Hidden markov models for optical flow analysis in crowds. In *Proceedings of the 18th International Conference on Pattern Recognition. ICPR 2006.*, volume 1, 2006.
- [AC07] N. Anjum and A. Cavallaro. Single camera calibration for trajectory-based behavior analysis. In *AVSS 2007, IEEE Conference on Advanced Video and Signal Based Surveillance*, pages 147–152, 2007.
- [BKS07] F.I. Bashir, A.A. Khokhar, and D. Schonfeld. Object Trajectory-Based Activity Classification and Recognition Using Hidden Markov Models. *IEEE Transactions on Image Processing*, 16:1912–1919, 2007.
- [DTS06] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. *Lecture Notes in Computer Science*, 3952:428, 2006.
- [EP09] Nathan Eagle and Alex Sandy Pentland. Eigenbehaviors: identifying structure in routine. *Behavioral Ecology and Sociobiology*, 63(7):1057–1066, 2009.
- [FGR03] G L Foresti, G Giacinto, and F Roli. Detecting dangerous Behaviors of Mobile Objects in Parking Areas. *Multisensor Surveillance Systems: The Fusion Perspective*, pages 199 FG – 0, 2003.
- [FMS04] G.L. Foresti, C. Micheloni, and L. Snidaro. Event classification for automatic visual-based surveillance of parking lots. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 3, pages 314–317. IEEE, 2004.
- [GJH] Aphrodite Galata, Neil Johnson, and David Hogg. Learning Variable-Length Markov Models of Behavior. *Computer Vision and Image Understanding*, 81(3):398–413.
- [Har75] J A Hartigan. *Clustering algorithms*. John Wiley & Sons, Inc., New York, 1975.
- [HEC06] D Hall, R Emonet, and J L Crowley. An automatic approach for parameter selection in self-adaptive tracking. *International Conference on Computer Vision Theory and Applications (VISAPP)*, Springer Verlag, Setúbal, Portugal, 2006.
- [Lap06] I Laptev. Improvements of object detection using boosted histograms. In *British Machine Vision Conference*, volume 3, pages 949–958, 2006.
- [LCSL07] I Laptev, B Caputo, C Schuldt, and T Lindeberg. Local velocity-adapted motion events for spatio-temporal recognition. *Computer Vision and Image Understanding*, 108:207–229, 2007.
- [LSW⁺06] F Lv, X Song, B Wu, VK Singh, and R Nevatia. Left luggage detection using bayesian inference. *Proceedings of the 9th IEEE International Workshop*, 2006.
- [MF03] C Micheloni and G L Foresti. Fast good features selection for wide area monitoring. *Proceedings. IEEE Conference on Advanced Video and Signal Based Surveillance, 2003.*, pages 271–276 FG – 0, 2003.
- [MG06] S. Munder and D.M. Gavrila. An Experimental Study on Pedestrian Classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:1863–1868, November 2006.
- [Pat11] *Incremental learning on trajectory clustering. In: Intelligent Paradigms in Safety and Security*, chapter 3. Springer-Verlag, 2011.

- [PC03] L. Panini and R. Cucchiara. A machine learning approach for human posture detection in domotics applications. In *12th International Conference on Image Analysis and Processing, 2003*, pages 103–108. IEEE Comput. Soc, 2003.
- [PFS05a] C. Piciarelli, G.L. Foresti, and L. Snidaro. Trajectory clustering and its applications for video surveillance. In *Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance, AVSS 2005.*, volume 18, pages 40–45. IEEE, 2005.
- [PFS05b] C. Piciarelli, G.L. Foresti, and L. Snidaro. Trajectory clustering and its applications for video surveillance. In *Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance, AVSS 2005.*, volume 18, pages 40–45. IEEE, 2005.
- [Por04] F. Porikli. Learning object trajectory patterns by spectral clustering. In *2004 IEEE International Conference on Multimedia and Expo (ICME)*, volume 2, pages 1171–1174. IEEE, 2004.
- [SG00] Chris Stauffer and W. Eric L. Grimson. Learning patterns of activity using real-time tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):747–757, 2000.
- [SP06] F. Scalzo and J.H. Piater. Unsupervised Learning of Dense Hierarchical Appearance Representations. In *18th International Conference on Pattern Recognition (ICPR'06)*, pages 395–398. Ieee, 2006.
- [SS05] Y Sheikh and M Shah. Bayesian modeling of dynamic scenes for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:1778–1792, 2005.
- [WB01] A D Wilson and A F Bobick. Hidden Markov models for modeling and recognizing gesture under variation. *International Journal of Pattern Recognition and Artificial Intelligence*, 15:123–160, 2001.
- [XG] Tao Xiang and Shaogang Gong. Incremental and adaptive abnormal behaviour detection. *Computer Vision and Image Understanding*, 111:59–73.
- [XG05] T Xiang and S Gong. Video behaviour profiling and abnormality detection without manual labelling. *Tenth IEEE International Conference on Computer Vision, 2005. ICCV 2005*, 2, 2005.
- [XGM05] Tao Xiang, Shaogang Gong, and Queen Mary. Relevance learning for spectral clustering with applications on image segmentation and video behaviour profiling. *Proceedings. IEEE Conference on Advanced Video and Signal Based Surveillance, 2005.*, pages 28–33, 2005.
- [ZGPBM05] D Zhang, D Gatica-Perez, S Bengio, and I McCowan. Semi-supervised adapted HMMs for unusual event detection. *IEEE Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005*, 1, 2005.