Camera Selection for Adaptive Human-Computer Interface

Niki Martinel *Student Member, IEEE*, Christian Micheloni, *Member, IEEE*,
Claudio Piciarelli, *Member, IEEE* and Gian Luca Foresti, *Senior Member, IEEE*

Abstract

Video analytics has become a very important topic in computer vision. This work introduces advanced Video Analytics Human-Computer Interfaces for a Video Surveillance System to ease the tasks of security operators. The visualization of the most relevant views is provided by the HCI module that pre-emptively activates cameras that will probably cover the motion of interesting objects. Human-Computer Interaction principles has been considered to develop the novel User Interface. Four prototypes have been designed and usability performance has been evaluated exploiting standard methods. Results obtained from such evaluations show the efficiency of the novel information visualization technique.

Index Terms

Video surveillance, Pre-emptive camera selection, Camera networks, Human-Computer Interface, Video analytics

1 Introduction

Video Surveillance Systems (VSSs) have rapidly progressed in the past 10 years [1]. Even though the number of cameras installed for surveillance purposes is increasing, it has been shown [2] that large scale deployments are still not supporting the requests since both low-level and high-level computer vision tasks are not enough robust yet.

Compared to the great amount of research done for the high level tasks [3], [4], [5], just a few researchers focused their attention on the usability of video analytics systems. Modern systems [6], [7], [8] still require operators' endeavor to monitor the vast amount of acquired data. As a result, the human attention and

- *N. Martinel, C. Micheloni, C. Piciarelli and G. L. Foresti are with the Department of Mathematics and Comptuer Science, University of Udine, 33100, Italy.*
  E-mail: {niki.martinel, christian.micheloni, claudio.piciarelli, gianluca.foresti}@uniud.it.

Manuscript received February 4, 2012; revised May 15, 2012; revised Jan 12, 2013; accepted May 15, 2013. This work is partially supported by the Interreg IV Italy-Austria project n. 4697 "SRSNet - Intelligent Audio/Video Sensor Networks".

capabilities are overpowered. Only in the last few years, the research community has proposed new user interfaces (UI) to better assist end-users in their monitoring tasks [8], [9], [10]. In particular, the new proposed methods for wide area analysis [11] highlight relevant areas and guide the user attention only on critical information while the development of UI for tracking tasks is almost not considered.

In current video analytics systems objects have to be followed through multiple cameras and surveillance operators have to switch between camera views and monitors as well. In many cases, to follow objects between camera views, video surveillance operators employ a single monitor which generally have quite small dimensions [12]. Commercial products usually propose VSSs that are equipped with huge wall screens and/or some remote smaller displays [13]. It is a matter of fact that such solutions still require a huge mental effort. For these reasons, VSSs must provide effective UIs such that relevant information are provided in a coherent and useful way.

The development of an effective and powerful information visualization technique is the goal of this work. The idea is to properly visualize only the most important cameras and information contents to simplify the operators' tasks. The main novelty is the dynamic organization, activation and switching of the UI elements based on the output of video analytics algorithms. Rather than displaying all available camera views, only most probable streams, i.e. those that will be involved with the objects motion, are presented. So, to reach the goal, two main challenges should be addressed: i) to distill the volumes of monitoring information into a human manageable quantity; ii) to present the filtered visual information to end-users such that they can take appropriate decisions in a limited amount of time.

The first challenge is addressed by the Video Analytics Module (VAM) using an approach similar to [14]. The hand-off between different camera views is used to track a single object among different fields-of-view (FoV) that are geographically adjacent. The proposed camera planning algorithm uses geographical clues and exploits the predicted trajectories to build an accurate camera activation plan. The camera activation plan together with the tracking data is used to provide only the most valuable data to the novel information visualization technique.

The Human-Computer Interface (HCI) addresses the second challenge. The new visualization algorithm exploits the VAM activation plan and tracking data to arrange UI elements accordingly to visual semantic information. In particular, camera views are arranged such that the operators have to focus only on relevant information. The proposed system uses the *overview plus detail* representation technique [15] to better display geographical clues.

The rest of the paper is organized as follows. A description of the system is given in section 3. Section 4 introduces the trajectory clustering algorithm and cluster trees. Details about the three main HCI components are given in section 5. Experimental results are shown in section 6. Finally, conclusions and future works are discussed in section 7.

2 RELATED WORK

The computer vision and video surveillance community have mainly focused on algorithms to extract valuable information from footages. Despite most of these algorithms are efficient and have high performance, the human part is still involved in the process of monitoring video streams from multiple cameras.

As pointed out in [12], the human ability to understand and interact with a large amount of data could be increased through visual analytic tools. A perceptual user interface that allows users interaction by means of gestures was introduced in [16]. In [8] an attention-aware human-machine interface (HMI) to monitor human operators attention was proposed. The VSAM project described in [17] demonstrates that a single human operator can effectively monitor a significant area of interest. The proposed UI exploits the VSAM technology to automatically display graphical representations of individuals into the digital environment. The ADVISOR system [18] selects relevant outputs and displays the relevant video feeds to the operator using a novel HCI. In [19] a framework for video surveillance based on the context of the experiential environment for efficient and adaptive computations was proposed. In [10] a Dynamic Object Tracking Systems introduced a novel VSS user interface. The same authors extended it by inspecting activity patterns [20] and introducing geometric tools [21]. A new backbone system that was used to develop advanced monitoring techniques, integrating cameras installed around the monitored area and centralized information, was introduced in [22]. In [9] a two-tiered VSS that self-adapts to current user needs was proposed. Similarly, in [23], the Virtual Document Planner was introduced to reduce the visual clutter and to display only situation-tailored information.

Similar techniques were proposed in commercial products. The IBM Smart Surveillance system (S3) [24], [25] uses a web-based service interface to support video based behavioral analysis. In [26] an integrated command and control solution designed to support security management is proposed. 3D site maps are displayed together with useful information to help contain and prevent dangerous events. Similarly, in [27] a graphical model of the monitored site allows users to select specific areas in order to display footages related to anomalous events. Finally, the Tag and Track system [28] allows users to select and track people across different camera views.

Despite many of these works help improving end-users capabilities, they still require huge mental efforts to the human operators. In particular, the main open issues are the followings: i) each user is required to monitor a large amount of footages at the same time; ii) tasks like tracking across multiple cameras require manual interaction with the UI to select desired camera views; iii) the position and the colors of UIs elements are not chosen accordingly to Human-Computer Interaction principles. The proposed work deals with those issues introducing: i) a predictive and autonomous selection of camera views; ii) a dynamic activation, selection and organization of video streams; iii) an information visualization technique that eases surveillance tasks. In Table 1,the properties of the proposed system are compared

TABLE 1
Comparison of the main properties of commercial and research systems with respect to the proposed one.

System	Wide Areas	Crowded Environments	On-line	Retrieve from repository	Multiple Object Tracking	Area Map	Multi-camera Visualization	Predictive Camera Selection
VSAM [17]	✔		✔		✔	✔		
ADVISOR [18]		✔	✔	✔	✔	✔		
DOTS [10], [20], [21]			✔			✔	✔	
IBM (S3) [25]	✔			✔			✔	
Siemens Surveillance Vantage [26]	✔		✔	✔		✔	✔	
Siemens SiteIQ [27]	✔		✔	✔		✔		
Ipsotek [28]	✔	✔	✔		✔	✔		
Proposed			✔		✔	✔	✔	✔

with the most important related works.

3 SYSTEM DESCRIPTION

As shown in Fig. 1, the architecture of the proposed VSS is organized in two main modules: i) the Video Analytics Module and ii) the Human-Computer Interface module.

The VAM module focuses on camera tasking operations. Video streams are analyzed to identify events of interest [29] that have to be provided to human operators together with useful information. For such a purpose, the VAM detects and recognizes all the active (i.e. moving or temporary stationary) objects in the monitored environment. When an object is acquired, the tracking algorithm starts to track the object. Then, a high-level component correlates the objects activities along time and space through the different camera views. Such an analysis is used by the trajectory estimator [30] to predict the trajectories of the objects of interest. By using past information about activities and trajectories, this component is able to path-plan the movements of the objects of interest such that the camera network can be opportunely tasked or redirected in order to improve the analysis capabilities [31]. The reconfiguration component proposed in [32] is used to automatically reconfigure the PTZ cameras and improve the system performance.

The estimated trajectories and the camera network configuration are input to the HCI module. The objective of the HCI module is to organize and display video streams to better support operators' tasks.

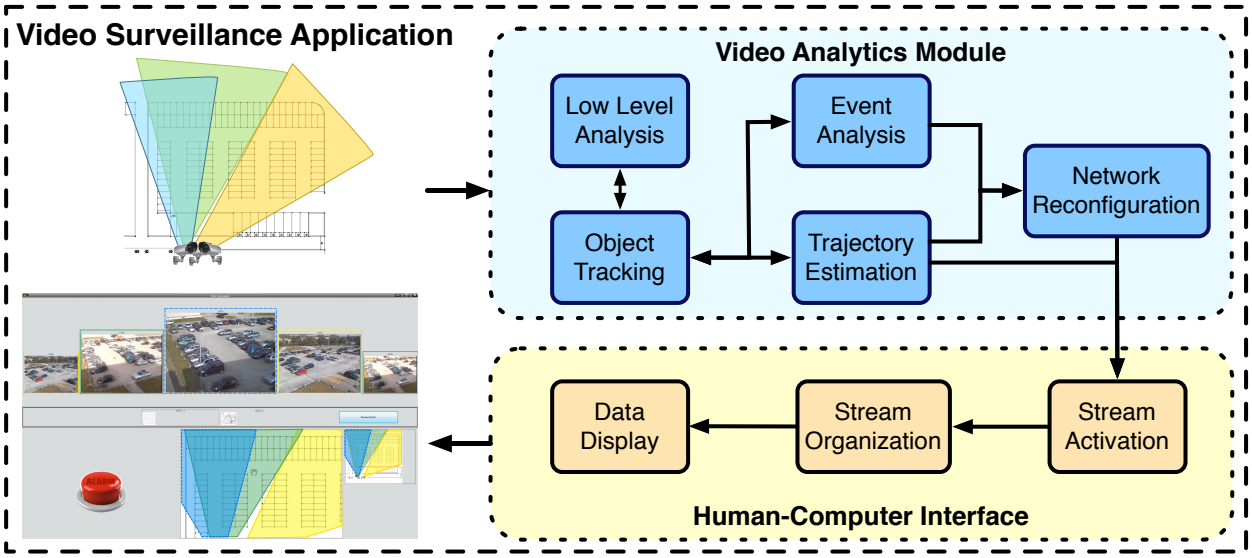


Fig. 1. Proposed system. The Video Analytics Module fuses information about trajectory predictions and object tracking to reconfigure the network and select only relevant streams. The HCI module organizes and displays the selected streams through an advanced UI.

The HCI module is composed by: i) the stream activation, ii) the stream organization and iii) the data display components. The stream activation component exploits VAM data to select and activate only relevant video streams. Given the estimated evolution of the environment (trajectories and involved cameras), it selects the streams that most probably will acquire objects activities. The stream organization sorts the selected camera views with respect to their estimated importance. Finally, the data display component displays the organized streams on the UI together with useful information provided by the VAM.

4 VAM MODULE

The Video Analytics Module extracts information about the events observed in the monitored environment by detecting moving objects and processing their trajectories. Moving objects are detected by means of a change detection algorithm and classified using a neural network, then the position of each object is filtered using a Kalman filter and a Camshift color tracker [29].

As new trajectories are acquired, the trajectory clustering algorithm proposed in [30] organizes the detected trajectory clusters in a probability-labeled tree. This allows to detect the clusters with higher probability of being matched, corresponding to the zones where it is more frequent to identify a moving object. This information is useful for event analysis tasks such as predicting object movements in the near future. The algorithm is here briefly summarized, for full details see [30].

4.1 Trajectory-cluster matching

A trajectory T_i is modeled by a list of vectors t_{ij}, each one representing the 2D spatial coordinates of object i at time j: $T_i = \{t_{i1} \dots t_{in}\}$ where $t_{ij} = (x_{ij}, y_{ij})$. The spatial coordinates can be computed directly on the image plane -even though in this work coordinates are expressed in a world reference frame. This is achieved by projecting the image plane position of each object on a map of the monitored environment using a homographic projection. Clusters (groups of trajectories with similar spatial features) are represented in a similar way, with the addition of an approximation of the local variance σ_{ij}^2 of the cluster i at time j: $C_i = \{c_{i1} \dots c_{in}\}$ where $c_{ij} = (x_{ij}, y_{ij}, \sigma_{ij}^2)$.

In order to check if a trajectory matches a given cluster, a trajectory-to-cluster distance has been defined. Given a trajectory $T = \{t_1 \dots t_n\}$ and a cluster $C = \{c_1 \dots c_m\}$ the adopted distance is defined as

$$D(T, C) = \frac{1}{n} \sum_{i=1}^{n} d(t_i, C) \tag{1}$$

where

$$d(t_i, C) = \min_j \left(\frac{dist(t_i, c_j)}{\sqrt{\sigma_j^2}} \right) \quad j \in \{\lfloor (1-\delta)i \rfloor \dots \lceil (1+\delta)i \rceil\} \tag{2}$$

with $\delta < 1$ constant and $dist(t_i, c_j)$ the Euclidean distance between the trajectory point t_i and the cluster point c_j omitting the variance component. Using equation 1 the distance of a trajectory from a cluster is thus the mean of the normalized distances of each trajectory point t_i with the closest cluster point within a temporal window whose size, controlled by parameter the δ, increases through time. The variable-size temporal window allows matching also in case of limited temporal shifts between trajectories and matching clusters, avoiding at the same time matches with excessively large temporal distances.

Finally, when a trajectory matches a cluster, the cluster itself must be updated with the information of the newly matched trajectory. The updating equations implement a running average with exponential forgetting of the trajectory data:

$$\begin{aligned} x &= (1-\alpha)x + \alpha\hat{x} \\ y &= (1-\alpha)y + \alpha\hat{y} \\ \sigma^2 &= (1-\alpha)\sigma^2 + \alpha[dist(t_i, c_j)]^2 \end{aligned} \tag{3}$$

where $c_j = (x, y, \sigma^2)$ and $t_i = (\hat{x}, \hat{y})$ are the matching points as in eq. 2.

4.2 Cluster trees

The trajectory-to-cluster matching and updating equations described in the previous section cannot be directly applied in real-life scenarios as typically only partial matches can be detected (e.g. a trajectory starts close to a cluster and later leaves it). In order to model these behaviors the concept of *cluster trees* is applied as in [30]. A cluster tree is a tree where each node is a cluster representing a spatial portion of the environments shared by a set of sub-trajectories, and arcs represent connections between clusters. For

example, the trajectories in Fig. 2(a) all share the same initial region, modeled by cluster c1. When the trajectories diverge toward two different regions, these regions are represented by two new clusters, c2 and c3, and their link with c1 is modeled in the tree structure shown in Fig. 2(c). The tree data structure is preferred to a graph one, since the system is forced to model as a single cluster only shared prefixes (initial parts of trajectories) rather than suffixes; this is most useful for trajectory prediction and anomaly detection tasks.

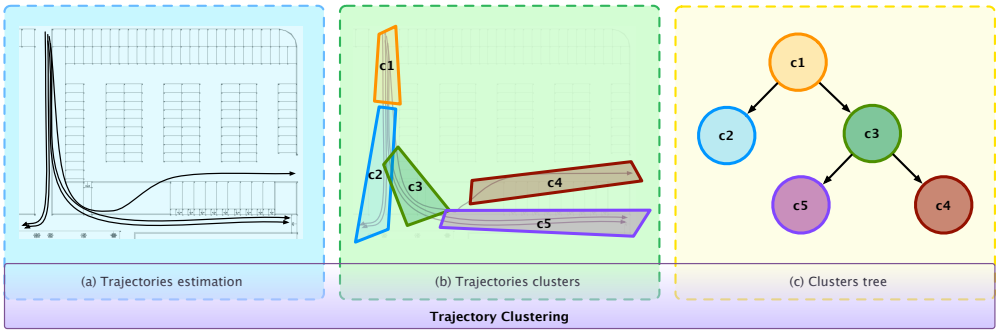


Fig. 2. Cluster trees represent the structure of a set of trajectories with partial sharing.

In order to create and update the cluster trees, the following procedure is used:

1) when a new trajectory is detected, its distance from existing clusters is computed using eq. 1;
2) if a match is not found, new cluster including the trajectory is created;
3) if a match is found, the cluster is dynamically updated according to eq. 3;
4) if the trajectory leaves a cluster:
   4a) the cluster is split in two parts in order to create a new branch if needed. The tree structure is updated accordingly;
   4b) a match among the children of the just-left cluster is searched, then the algorithm is iterated from point 2).

Points 2) and 4) rely on the trajectory-to-cluster distance $D(T, C)$ defined in equation 1 to check if a trajectory is matching or leaving a cluster. The distance is normalized according to the local cluster variance, and thus it is directly linked to the probability of the trajectory to belong to the statistical model represented by the considered cluster. For example, if $D(T, C) < 2$, it means that on average the trajectory falls within the 2σ range from the cluster center (a range including the 95% of the trajectories represented by that statistical model).

The described procedure allows to dynamically create and update cluster trees such as the one shown in Fig. 2(c). Arcs can be labeled with probabilities, computed by counting the number of trajectories matching each node. Specifically, if node C has n children nodes $c_1 \dots c_n$, the arc connecting C and c_i is labeled with probability $\frac{|c_i|}{\sum_{j=1}^{n} |c_j|}$ where $|c_i|$ is the number of trajectories matching cluster c_i.

In [30], labelled arcs are used for anomaly detection. The total probability of a fully developed trajectory is defined as the product of all the probabilities in the path from the first to the last node matched by the trajectory. Probabilities are used to predict the most probable future evolution of a partial trajectory. This feature is exploited in the proposed work to automatically select and organize the cameras that most probably will observe a given object.

5 HCI MODULE

Information about sensors streams and objects trajectories extracted by the VAM module are used by the HCI module to tailor contents that have to be displayed to the end-users. Three innovative components are introduced by the HCI module: i) the stream activation, ii) the stream organization and ii) the data display.

5.1 Stream activation

The stream activation component connects information given by the VAM to the stream organization and data display components. It uses the information from the trajectory estimation and network reconfiguration components to select and activate only relevant streams. In particular, the estimated path correlated to the fields-of-view computed by the network reconfiguration component, allows to plan the hand-off and activate the cameras that will, most probably, cover the motion of the object. Such cameras are then included in a priority queue that is used to keep the visual focus on the selected object.

Let Q be the priority queue, cam_j be the j-th camera with FOV_j field of view, then cam_j is pushed in Q if $FOV_j \cap T_i \neq \emptyset$, where T_i is a predicted trajectory.

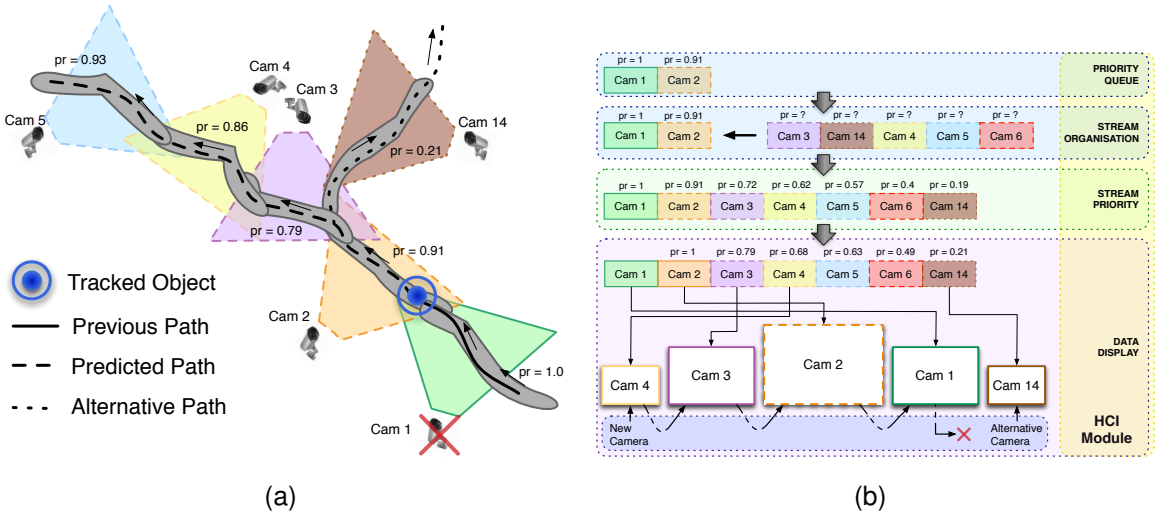
5.2 Stream organization

The stream organization component organizes camera views such that only the most relevant views are presented to the end-users. As shown in Fig. 3, the component achieves its objective re-weighting the streams that have been inserted into the priority queue and sorting the camera views accordingly to their estimated importance.

Streams that have been previously inserted into the priority queue by the stream activation component are evaluated against all the possible object trajectories taking into account the geographical deployment of sensors. Thus, according to the most probable trajectories given by the VAM module, the stream organization component assigns to each view a priority value computed by intersecting the trajectory clusters with each camera FoV that has been inserted into the priority queue.

The stream priority value is computed by traversing the predicted path tree (see Fig. 2). The edge value $P(C_i|C_j)$, connecting the clusters C_i and C_j, represents the probability of the object to reach cluster C_i given its previous position in cluster C_j. Hence

$$P(C_i|C_j, C_{j-1} \ldots, C_k) = P(C_i|C_j) \prod_{l=j}^{k+1} P(C_l|C_{l-1}) \tag{4}$$



(a) (b)

Fig. 3. In (a) the tracked object and the predicted path are shown together with camera FoV. In (b) the corresponding behavior of the HCI module components is shown.

is the probability that the object will reach the cluster C_i through the path $C_i, C_j, C_{j-1}, \ldots, C_k$. Thus, the camera in the queue that covers the cluster C_i is assigned with a priority value equal to $P(C_i|C_j, C_{j-1} \ldots, C_k)$. The camera covering the cluster where the object is currently in is assigned a priority of 1. Once the priority values have been computed, the queue is sorted in order to have higher priority cameras on top.

5.3 Data display

The data display component introduces a novel information visualization technique that aims to ease surveillance operators tasks exploiting Human-Computer Interaction principles. As Fig. 4 shows, the proposed UI introduces two main components: i) the video streams area and ii) the map area. The video stream area organizes and displays the camera view UI elements inserted in the priority queue to better support end-users tasks. The "switch panel" allows to switch between different areas of the monitored environment and to organize the video streams on the basis of the objects of interest. In case of multiple objects of interest, the operator is able to follow one of them just by switching the active visualization. The map of the area displays geographical information about cameras positions, cameras FoV and moving objects.

5.3.1 Video streams area

The video streams area introduces a novel information visualization technique to display only the most relevant views. Three main novel features are introduced by this component: i) camera views displacement; ii) camera views animation; iii) camera views representation.

The *camera view displacement* is organized such that, the stream of the sensor with highest priority is displayed at the center of the video streams area (see Fig. 3). The streams that have lower priority values

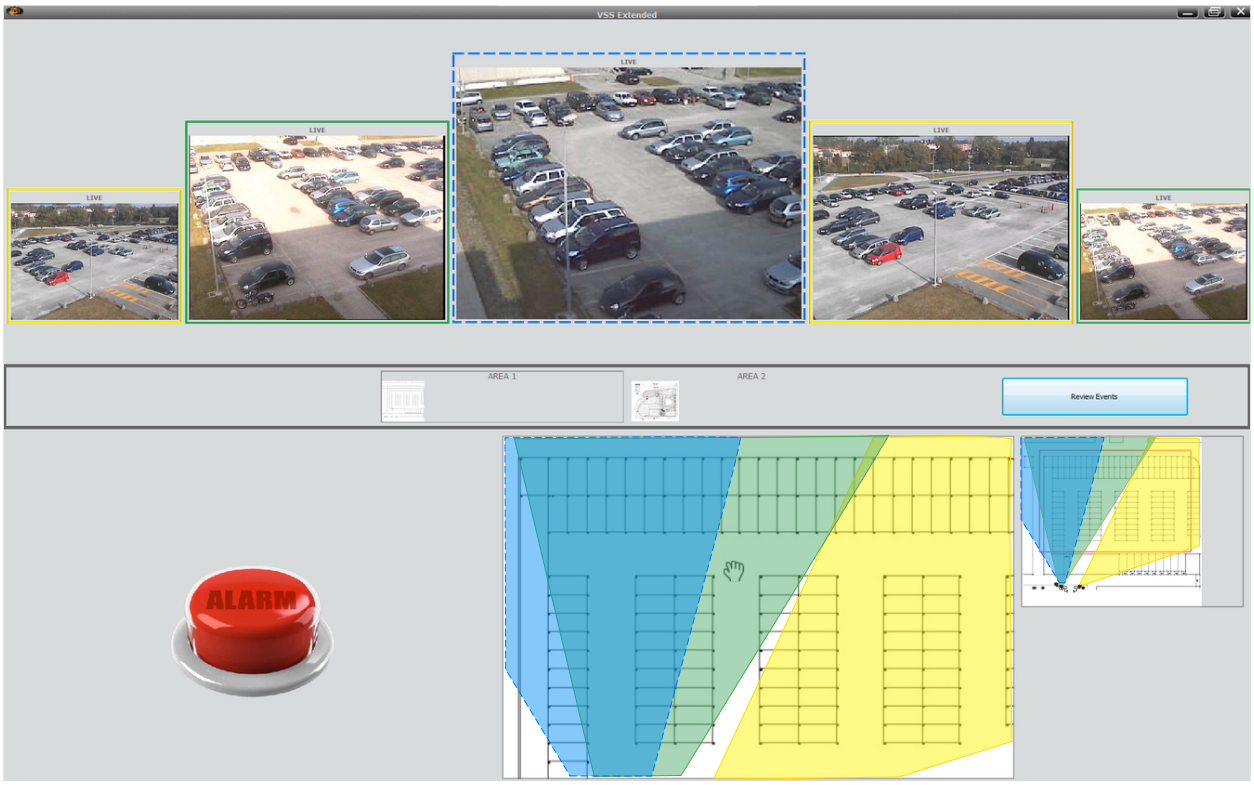


Fig. 4. Finally proposed User Interface. The top region shows five camera views that are displayed accordingly to the priority queue computed by the VAM. The bottom region shows the map area component together with the active camera fields-of-view.

are displayed either at the left or a the right side -depending on the predicted path of the object- of the main camera view. The previous highest-priority camera is shown on the other side. The goal is to provide the operator the previous and the next camera views that cover the predicted object trajectory.

It could be possible that the object of interest does not follow the most probable estimated trajectory, so, the most probable alternative path is considered. The camera view that has been assigned the highest priority with respect to such alternative path is displayed next to the previous highest-priority camera view (see Fig. 3).

For instance, let consider Fig. 3 and let assume that the tracked object is moving -from right to left- along the predicted path. Since the object is moving from Cam2 towards Cam3, Cam3 is displayed at the left of the current main view. As Cam2 is the highest priority camera, Cam1 is displayed at its right. The alternative camera with highest priority, Cam14, is placed next to Cam1. The camera priority is also used to set the size of the displayed camera views. The camera view with the highest priority has the largest size. Other camera views are scaled to $2/3$ of the size of the camera view with a higher priority.

The *camera views animation* is introduced to smooth the hand-off between camera views. Accordingly to

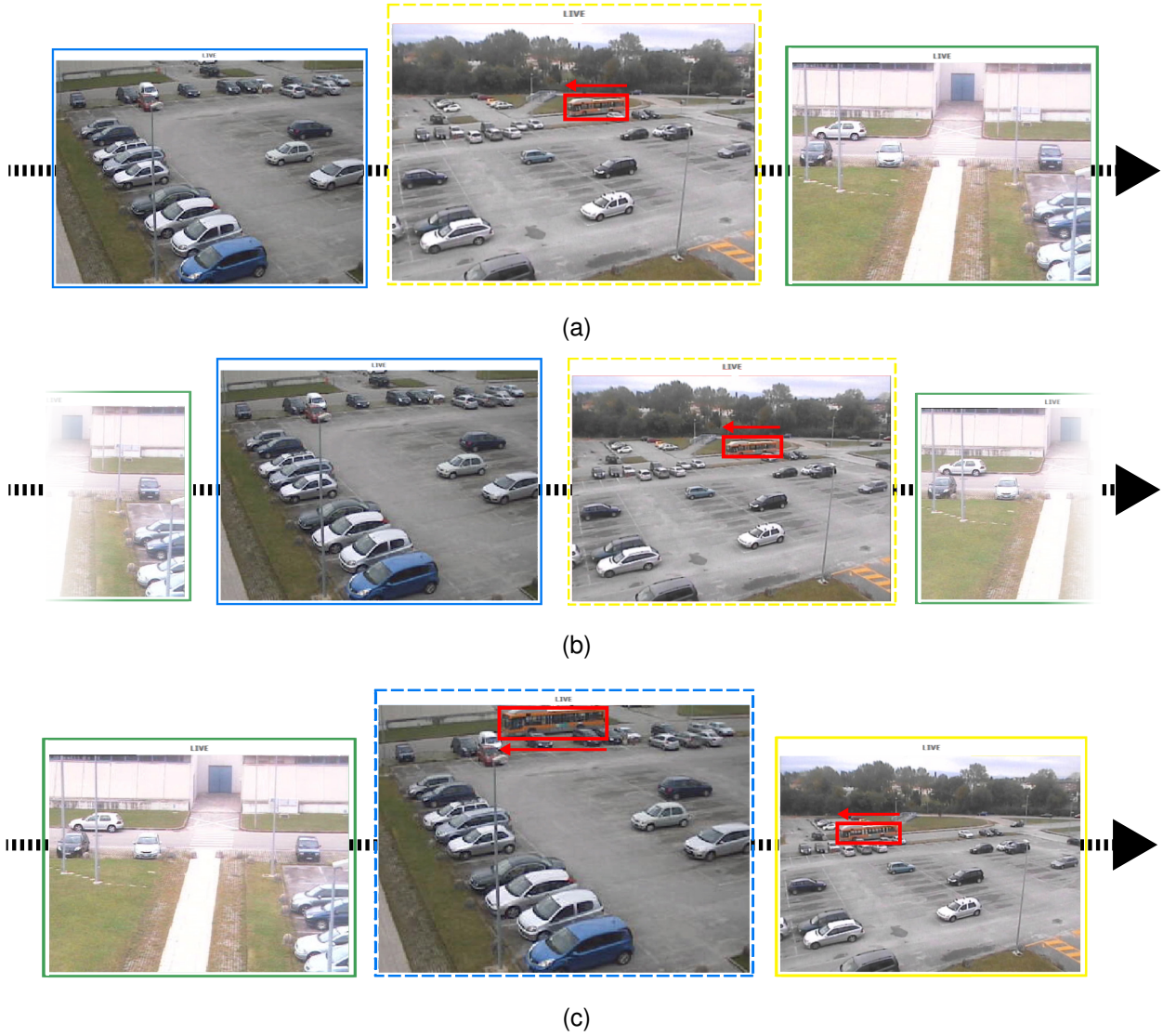
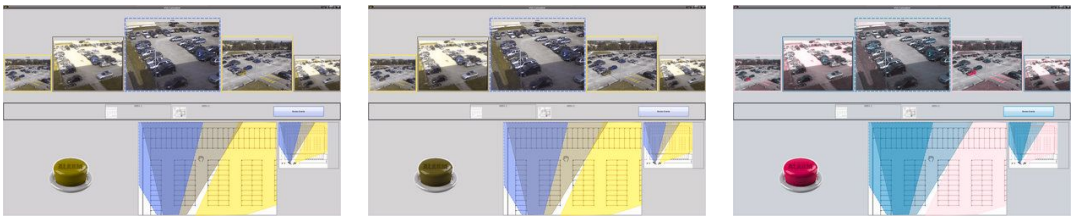


Fig. 5. Proposed camera views transition. The tracked object is moving from right to left. Camera views are scaling and moving to the right. In (a) the initial camera view UI element position is shown. In (b) the scaling and transition of all the camera views to the right is depicted. Finally, in (c) the updated camera view UI element position after one transition is shown.

the camera view displacement feature, the stream of the sensor with the highest priority is displayed at the center of the video streams area. Since objects are moving across a path, the camera views have to be moved to respect the stated objective. If camera views are just switched a "flashing" effect is introduced due to the fact that the relevant streams will be activated/deactivated at different camera views positions. Such behavior cause confusion to the end-users and it does not respect Human-Computer Interaction principles. To sidestep this issue, UI animation effects are introduced.

As shown in Fig. 5, as long as the object of interest follows the predicted trajectory, the relevant camera views are moved to the opposite direction with respect to the predicted object trajectory. Given the homography transformation between camera views and the map of the monitored environment (see section 4.1), the data display component estimates the velocity of the tracked object at each time instant and moves the UI camera views accordingly. Similarly, as camera views move, they are gradually scaled to the new sizes. Old selected views are scaled down and animated out of the UI.



(a) (b) (c)

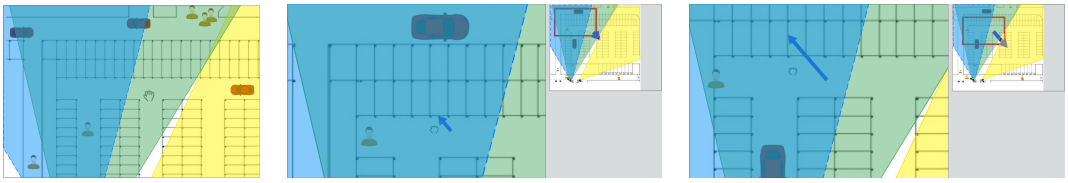
Fig. 6. Color-blind people vision simulation of the proposed UI: (a) Deuteranope simulation (red/green color deficit) (b) Protanope simulation (red/green color deficit) (c) Tritanope simulation (blue/yellow deficit)

The *camera views representation* introduces two main representation features: i) the color-coded and ii) the drawing style techniques used to depict the camera view UI elements. To achieve internal coherence the same representation techniques are used to depict the camera FoV in the map area component. The colors used to depict UI elements have been selected such that each camera view can be distinguished even from colour-blind people (see Fig. 6). To ease the end-users tasks, a different drawing style has been used to highlight the most relevant view. A dashed line is used to represent the camera with the highest priority. This allows to easily link the main camera view representation in the video streams area with its FoV depicted in the map area.

5.3.2 Map area

The map area introduces a component that shows the topological representation of the monitored area. As shown in [10], the map representation of the monitored area improves the ability to follow objects and to analyze their behavior while these are moving across different camera views. Similarly to state-of-the-art video analytics systems, cameras, FoV and objects are represented in the proposed map area (see Fig. 7). In addition to that, the work introduces a novel map visualization technique. Though the standard scrolling, panning and zooming techniques are useful to explore an information space at different levels of detail, it is often useful to display more than one level of detail at the same time [33]. The *overview plus detail* technique [15] is exploited to achieve such objective. This technique helps users to keep focusing on the details of an information space without loosing the overview of the entire space.

The *overview plus detail* technique is used to display both the detailed map and the context view. The context view displays a downscaled version of the map. It also highlights the portion of the map



(a) (b) (c)

Fig. 7. Proposed map area UI component. In (a) a standard map representation together with objects positions is shown. In (b) and (c) the overview plus detail representation is shown. Both the overview and the detail view can be zoomed and panned. The viewfinder (red box) is updated accordingly.

displayed in the detail view with a rectangular viewfinder. Both the viewfinder and the detailed view can be dragged to navigate the environment. The size and the position of the viewfinder in the context view also provide useful information for navigation, such as details about the scale between the displayed detailed map portion and the whole map. Thanks to the novel visualization technique, the operator has an overview of the entire area even if it has zoomed the map view to retrieve more details about an object (see Fig. 7(b) and Fig. 7(b)).

6 Experimental Results

Human-Computer Interaction guidelines have been followed to evaluate the usability performance of the novel system. Four prototypes have been designed respecting the main usability rules defined in [34]. Empirical and non-empirical methods have been employed to evaluate each prototype.

To correctly apply the Human-Computer Interaction principles, information about classes of users and context of use have been identified. Users have been grouped in the following four different classes:

1) operators that use a VSS to monitor a small area for private purposes;
2) operators that use a VSS to monitor a small public environment;
3) operators that use a VSS with a multi-camera setup to monitor a wide area;
4) operators that use a VSS with a multi-camera setup to monitor multiple wide areas.

The second step was to identify the most probable contexts of use of the system. Five contexts of use have been identified:

1) visualize video streams using single-multiple displays;
2) track objects through multiple cameras FoV;
3) fire alarms;
4) automatic recording of interesting events footages;
5) review recorded footages.

All the four proposed prototypes have been validated using empirical and non-empirical tests. The following six different evaluation tasks have been proposed:

1) visualize the real time footage from "AREA 2";
2) fire the alarm if an anomalous event occurs;
3) associate current visible streams to area sensors;
4) start tracking an object and follow it along its path;
5) start tracking an object, then fire the alarm if an anomalous event occur;
6) start tracking an object, then switch to a different area and start tracking a new object.

Non-empirical evaluations have been performed with the support of four Human-Computer Interaction experts. The non-empirical techniques have been used to obtain an initial evaluation of each prototype. After the evaluation, each prototype has been reviewed accordingly to the reports provided by the HCI experts.

To evaluate the prototypes, the six stated tasks have been performed by the experts. The steps required to reach each given task have been analyzed using two common techniques: i) the heuristic evaluation [35] and ii) the cognitive walkthrough [36]. After the evaluation, the experts provided a review for each of the 10 principles proposed in [37].

The cognitive walkthrough technique has been used to mainly detect the UI design errors that affected the ease of learning. A review for each UI feature, behavior and action involved in the proposed tasks has been given by each HCI expert.

All the recommendations provided by HCI experts have been taken into account to solve the identified problems. The process strongly helps the design and lets the system to perform better in terms of affordance, visibility and coherence with respect to standard video surveillance system UIs.

Empirical evaluations have also been performed to validate the proposed prototypes. Three standard empirical evaluation methods have been used to evaluate the usability performance: i) thinking aloud; ii) video screen recording and iii) usability questionnaires. To perform the empirical evaluations forty pre-identified end-users have been selected (see Tab.2) and grouped into the four proposed clusters. As for non-empirical evaluation, they have been asked to perform the same six evaluation tasks.

		Years of experience			
		0-1	2-5	5-10	10+
Real Operators	Male	2	4	5	2
	Female	2	4	0	0
Others	Male	4	5	2	1
	Female	6	2	1	0

TABLE 2

Forty pre-identified users have been selected to evaluate the performance of the proposed prototypes.

The test sessions have been conducted in a controlled environment using pre-recorded video data.

During such sessions users were supported by the researcher that was not allowed to intervene unless the end-user were not able to achieve the goal or if they had some questions about the UI elements behavior that did not reflect their expectations. Video screen recordings have been captured during test sessions. After completing all the assigned tasks, the usability questionnaires have been given to each user. All the acquired data has been merged and compared to detect and solve the prototypes issues.

To get a quantitative evaluation of each designed UI two indexes have been proposed: i) the mean success rate index and ii) the mean execution time index. Let $p = \{1, 2, \ldots, P\}$ be the proposed interface and let $j = \{1, 2, \ldots, J\}$ be the current evaluation task. The mean success rate index (MSR) provides information about how well p scales to j. It is given by:

$$MSR(p, j) = \frac{\sum_{i=0}^{n_u} C_{i,j}^{p}}{n_u} \tag{5}$$

where $C_{i,j}^{p}$ is a matrix that gives the completion percentage of task j reached by user i using prototype p. n_u is the total number of tests. This index has been evaluated against each single task j and each proposed prototype p in order to see if the current solution improves the previous one.

Similarly to the MSR index the mean execution time index (MET) has been computed to investigate the UI efficiency. The MET is computed as

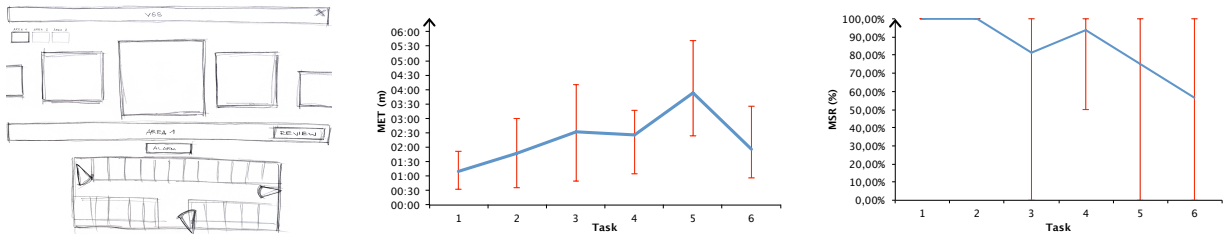
$$MET(p, j) = \frac{\sum_{i=0}^{n_u} T_{i,j}^{p}}{n_u} \tag{6}$$

where $T_{i,j}^{p}$ is a matrix that gives the time required by user i to complete task j using prototype p. In case a user was not able to fully complete the required task, the time assigned to that user is given by $\max T_{k,j}^{p}$ with $k \neq i$. The MET index has been used to provide information about how much time a single user needs to reach a given goal (user failure has been taken into account as well). By analyzing this data, it was possible to identify which were the tasks that required more time. In particular, during the design process, if a given task was requiring too much time the UI elements involved in that process were deeply inspected before evaluating the next prototype.

In all the experiments, the distance threshold required by the clustering algorithm (see section 4.2) has been empirically fixed to 2. Since the trajectory-to-cluster distance is normalized by the cluster variance, this means that a trajectory matches a cluster if, on average, its distance from the cluster center lies in the 2σ range.

6.1 Evaluation of the first prototype

A paper model has been used to design the first UI prototype. As defined by HCI rules, this is a common solution that allows a faster and easier definition of the system UI. As shown in Fig. 8(a), the model does not introduce any color that allows people to identify cameras and to relate their FoV. This choice allowed to investigate how people associate cameras views and their UI map representation. The task #3 is thus hard to perform under this scenario, and, as results depicted in Fig. 8(b) and Fig. 8(c) show, some users



(a) (b) (c)

Fig. 8. First system prototype (paper). The first model lacked of colors and had poor interaction but was equally helpful to detect the initial design issues. (a) Proposed user interface, (b) Mean Execution Time and (c) Mean Success Rate.

did not complete it. The MSR for this specific task is about 81%, and the standard deviation is about 37%.

Even though the proposed UI was completely static, some of the given tasks required to track objects. To sidestep this issue the UI elements behavior has been simulated by moving paper objects. Mainly because of that, users failed to perform actions that required non-static UI elements and live video streams. In particular, task #5 required a higher amount of time to be completed since the anomalous events were displayed at 0'30", 1'14", 2'11', 3'38' and 5'40". Notice that events used in task #2 were not the same as those used in task #5. Results reported in Fig. 8(b) shows that all the users had trouble with such task.

As shown in Fig. 8(c), users failed to reach the required goal for task #3, #5, and #6. In particular, task #6 has the lowest MSR (about 56%).

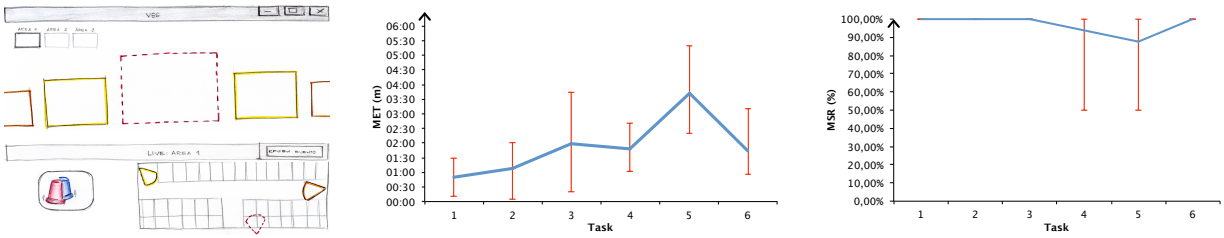
The main problems that came from the evaluations of the first prototype were:

- the lack of colors and techniques that allowed users to relate cameras depicted on the map with the camera views in the video streams area;
- the lack of video streams and the low interaction;
- the lack of interaction with the map.

6.2 Evaluation of the second prototype

As for the previous prototype, a paper model has been used to design the second UI prototype. To solve the issue detected from the previous evaluation two main novelties have been introduced: i) usage of colors and ii) change of the *alarm* text-button to an icon-button.

As Fig. 9(a) shows UI colors have been added. The MET index (see Fig. 9(b)) shows that such feature did not significantly decrease the average time required to perform the proposed tasks. In particular, as Fig. 8(b) and Fig. 9(b) highlight, the task #3 reached a mean execution time of about 2'32" for the first prototype and an average time of 2'01" for this second prototype. The other tasks, if compared to the first prototype, achieved similar MET results even the standard deviation for each of them is about 6%



(a) (b) (c)

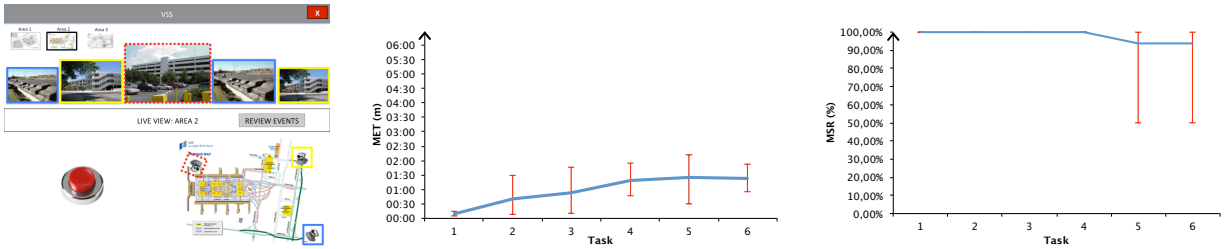
Fig. 9. Second system prototype (paper). The second model introduced the colors different depiction techniques to associate the camera views in the video stream area and the cameras in the map area. (a) Proposed user interface, (b) Mean Execution Time and (c) Mean Success Rate.

lower. As for the previous evaluation task #5 was the one that required much time to be performed since anomalous events were shown at 0'30", 1'14", 2'11', 3'38' and 5'40". As before, events used in task #2 were not the same as those used in task #5.

Though the MET index doesn't show any significant improvement by the new prototype, the MSR index shows that the proposed colors and the employed depiction techniques solved the previously detected issues. In particular, the mean success rate for the task #3 had strongly increased from about 81.25% to 100%. The same happened for task #6. In both cases, all the users achieved the required tasks reaching a 100% MSR score. But, as shown in Fig. 9(b) and Fig. 9(c), task #4 and task #5 were still difficult to perform and required a long time to be completed.

Similarly to the first prototype evaluation, the main issues posed by this second prototype were:

- the lack of video streams and the low interaction;
- the behavior and the representation of the *alarm* icon-button;
- the lack of interaction with the map.



(a) (b) (c)

Fig. 10. Third system prototype (interactive model). The third model introduced video streams and an interactive map. (a) Proposed user interface, (b) Mean Execution Time and (c) Mean Success Rate.

6.3 Evaluation of the third prototype

As Fig. 10(a) shows a more interactive model has been used to design the third prototype. The third prototype has been developed using a presentation software. The slides of the presentation were arranged to simulate a real software. Footages recorded from a real-surveillance scenario have also been added. In order to allow users to perform all tasks, behaviors of UI elements involved by the required tasks have been defined. The main novelties introduced by the third prototype were: i) higher-level interactions and ii) change of the *alarm* button.

Similarly to the previous evaluations, both the MET and the MSR indexes have been computed. As the MET index shows (see Fig. 10(b)) the amount of time required to perform each single task has decreased with respect to the two previously proposed prototypes. The standard deviation -of the MET index- computed for all the six given tasks has averagely decreased of about 64%. The amount of time required to complete the task #1 was about 0'15". Similarly, the MET for task #3 has decreased from 2'01" (second prototype) to 0'48". The strongest improvement has been achieved by the task #5, where the MET has decreased from 3'41" to 1'20". Since the same anomalous events have been used (as in the first and the second prototype evaluation), the achieved results show that most of the users missed only the first anomalous event (at 0'30").

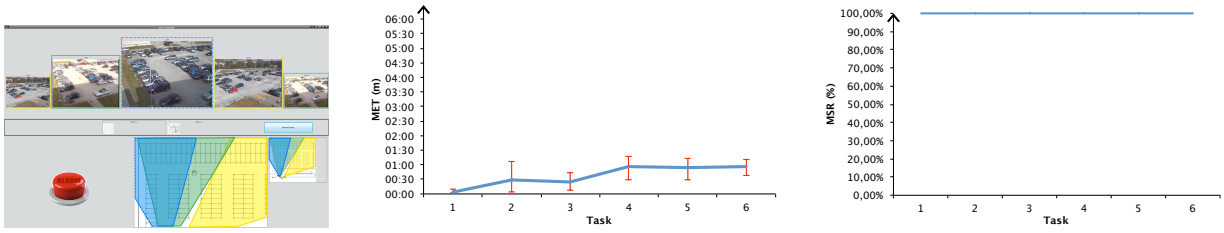
The MSR index shows similar results, to the ones achieved by the second prototype, for task #1, task #2, and task #3. A 100% MSR has been reached by task #4. In contrast with results obtained from the previous evaluation, task #6 was not fully completed by all users (see Fig.10(c)). A MSR score of 93% has been achieved by both task #5 and #6. The problem was that the *alarm* icon-button was hard to understand and its behavior was not clear. Users also expected to use the map to select the objects to track.

Results of end-users tests conducted among the third prototype showed that the proposed UI elements had a better affordance, but some issues were still present. Single-user questionnaires inspection and results analysis showed that the main negative aspects posed by the third prototype were:

- when the active, the *alarm* button showed visual clues but no sound information was provided;
- the *alarm* button was misunderstood by many users;
- the lack of interaction with video streams. The selected videos came with multiple objects and some users expected to start tracking a chosen object by clicking through it. The prototype was not designed to allow such interaction and it started tracking a different object with respect to the selected one.

6.4 Evaluation of the fourth prototype

A software program has been developed as the fourth prototype (see Fig. 11(a)). As for other prototypes, the same prerecorded data has been used. The main novelties introduced by such prototype were: i) the *overview plus detail* UI element, ii) the depicted cameras FoV and iii) the representation of objects within the map.



(a) (b) (c)

Fig. 11. Fourth system prototype (software). Both the VAM and the HCI modules were designed and the same prerecorded data has been used to evaluate the performance of the system prototype. (a) Proposed user interface, (b) Mean Execution Time and (c) Mean Success Rate.

As Fig. 11(b) shows the MET index decreased with respect to the MET computed for the third prototype. Also, the MET standard deviation computed for all the six given tasks decreased of about 47% on average. The strongest improvements have been achieved by the task #5 and task #6. In contrast with previous evaluations, task #5 wasn't the one that required the longest time to be performed: all the users catch the first anomalous event. The higher MET was achieved by the task #6 since many users had difficulties in selecting an object to track by clicking through it on the video streams UI representations. The MET required to perform each single task was always less than a minute.

The best results come from the MSR index since all the tasks achieved a 100% score. Such results show that the depicted camera FoV and the representations of the object onto the map component ease the surveillance tasks. None of the real-operators nor the novel-users asked to view all other camera streams that were not shown in the given UI. This is a very interesting result if compared to standard VSS UIs where users have to manually switch between camera views to activate them to follow an object of interest.

7 Conclusions and Future Works

In this paper, a novel information visualization technique for Video Surveillance Systems has been introduced. The video analytics system introduces the VAM and the HCI modules to properly visualize only the most important cameras and information contents, thus simplifying surveillance tasks.

The VAM performs video analytics tasks and predicts the possible paths of the objects of interest. Trajectories and cluster trees learned from real-tracking data are used to predict the most probable paths of tracked objects.

The HCI module presents only relevant information to surveillance operators selecting the streams accordingly to information provided by the VAM module. It introduces three main components to propose a novel information visualization technique for VSS.

Four UI prototypes have been designed and evaluated using standard Human-Computer Interaction techniques. Non-empirical evaluations results have been fused together with two proposed indexes to detect and solve usability issues introduced by each prototype. The results shows that the adopted information visualization technique achieves high usability results and supports end-users during their surveillance tasks.

The following three main problems affect the current system. i) The system can be used in situations where the monitored scenario is not overcrowded. ii) In case the size of the display is small, the camera views displayed in the video stream area may be too small and the task of recognizing objects may be hard. iii) If the number of objects to track is very high, the "switching panel" gets overcrowded and users may get confused by that. To address those issues, robust techniques that allows object tracking over crowded environments and non-overlapping cameras [38] will be introduced. New displacement methods to better display the camera views in the video stream area and the "switch panel" will be analyzed as well. In addition, Video analytics systems for wide area surveillance [11] will be investigated to integrate Unmanned Aerial Vehicle videos streams. A dynamic hierarchical view would be investigated to allow operators to select the number of cameras that have to be displayed for a particular area and for a particular task.

REFERENCES

[1] H. M. Dee and S. A. Velastin, "How Close Are We to Solving the Problem of Automated Visual Surveillance? A Review of Real-world Surveillance, Scientific Progress and Evaluative Mechanisms," *Machine Vision and Applications*, vol. 19, no. 5-6, pp. 329–343, May 2007.

[2] G. Foresti, C. Micheloni, L. Snidaro, P. Remagnino, and T. Ellis, "Active video-based surveillance system: the low-level image and video processing techniques needed for implementation," *IEEE Signal Process. Mag.*, vol. 22, no. 2, pp. 25–37, Mar. 2005.

[3] L. Lee, R. Romano, and G. Stein, "Monitoring activities from multiple video streams: establishing a common coordinate frame," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 758–767, Aug. 2000.

[4] C. Micheloni, P. Remagnino, H.-L. Eng, and J. Geng, "Intelligent Monitoring of Complex Environments," *IEEE Intelligent Systems*, vol. 25, no. 3, pp. 12–14, May 2010.

[5] P. Turaga, R. Chellappa, V. Subrahmanian, and O. Udrea, "Machine Recognition of Human Activities: A Survey," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 11, pp. 1473–1488, Nov. 2008.

[6] A. Hampapur, R. Bobbitt, L. Brown, M. Desimone, R. Feris, R. Kjeldsen, M. Lu, C. Mercier, C. Milite, S. Russo, C.-f. Shu, and Y. Zhai, "Video Analytics in Urban Environments," in *International Conference on Advanced Video and Signal Based Surveillance*. Genova, IT: IEEE, Sep. 2009, pp. 128–133.

[7] P. H. Tu, G. W. Brooksby, G. Doretto, D. W. Hamilton, N. Krahnstoever, J. B. Laflan, X. Liu, K. A. Patwardhan, T. Sebastian, Y. Tong, J. Tu, F. W. Wheeler, C. M. Wynnyk, Y. Yao, and T. Yu, "Video Analytics for Force Protection," in *Distributed Video Sensor Networks*, 1st ed., B. Bhanu, C. V. Ravishankar, A. K. Roy-Chowdhury, H. Aghajan, and D. Terzopoulos, Eds. London: Springer London, 2011, ch. 27, pp. 408–425.

[8] X. S. Zheng, J. Kiekebosch, and R. Rauschenberger, "Attention-aware Human-Machine Interface to Support Video Surveillance Task," in *Human Factors and Ergonomics Society Annual Meeting*, Princeton, NJ, Sep. 2011, pp. 1818–1822.

[9] P. Bottoni, M. D. Marsico, S. Levialdi, G. Ottieri, M. Pierro, and D. Quaresima, "A Dynamic Environment for Video Surveillance," *Human-Computer Interaction INTERACT 2009*, vol. 5727, pp. 892–895, 2009.

[10] A. Girgensohn, T. Dunnigan, D. Kimber, J. Vaughan, T. Yang, F. Shipman, T. Turner, E. Rieffel, L. Wilcox, and F. Chen, "DOTS: Support for Effective Video Surveillance," in *International Conference on Multimedia*. New York, New York, USA: ACM Press, Sep. 2007, p. 423.

[11] A. K. Roy-Chowdhury and B. Song, *Camera Networks: The Acquisition and Analysis of Videos over Wide Areas*, Jan. 2012, vol. 3, no. 1.

[12] G. Robertson, D. Ebert, S. Eick, D. Keim, and K. Joy, "Scale and complexity in visual analytics," *Information Visualization*, vol. 8, no. 4, pp. 247–253, Jan. 2009.

[13] S. Velastin, "CCTV Video Analytics: Recent Advances and Limitations," in *Visual Informatics: Bridging Research and Practice*, H. Badioze Zaman, P. Robinson, M. Petrou, P. Olivier, H. Schröder, and T. K. Shih, Eds., 2009, vol. 5857, pp. 22–34.

[14] F. Z. Qureshi and D. Terzopoulos, "Planning Ahead for PTZ Camera Assignment and Handoff," in *International Conference on Distributed Smart Cameras*, Como, Italy, 2009, pp. 1–8.

[15] A. Cockburn, A. Karlson, and B. B. Bederson, "A review of overview+detail, zooming, and focus+context interfaces," *ACM Computing Surveys*, vol. 41, no. 1, pp. 1–31, Dec. 2008.

[16] G. Iannizzotto, C. Costanzo, F. La Rosa, and P. Lanzafame, "A multimodal perceptual user interface for video-surveillance environments," in *International Conference on Multimodal Interfaces*. Trento: ACM Press, 2005, pp. 45–52.

[17] R. T. Collins, A. J. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, O. Hasegawa, P. Burt, and L. Wixson, "A System for Video Surveillance and Monitoring," Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. 4, Dec. 2000.

[18] N. T. Siebel and S. J. Maybank, "The ADVISOR Visual Surveillance System," in *Workshop Applications of Computer Vision*, Prague, CZ, 2004, pp. 103–111.

[19] J. Wang, M. S. Kankanhalli, W. Yan, and R. Jain, "Experiential Sampling for video surveillance," in *SIGMM International Workshop on Video Surveillance*. New York, New York, USA: ACM Press, 2003, p. 77.

[20] A. Girgensohn, F. Shipman, and L. Wilcox, "Determining activity patterns in retail spaces through video analysis," in *International Conference on Multimedia*. Vancouver, British Columbia, Canada: ACM Press, 2008, pp. 889–892.

[21] E. G. Rieffel, A. Girgensohn, D. Kimber, T. Chen, and Q. Liu, "Geometric Tools for Multicamera Surveillance Systems," in *IEEE International Conference on Distributed Smart Cameras*, ACM/IEEE. IEEE, Sep. 2007, pp. 132–139.

[22] B. T. Morris and M. M. Trivedi, "Contextual Activity Visualization from Long-Term Video Observations," *IEEE Intell. Syst.*, vol. 25, no. 3, pp. 50–62, May 2010.

[23] N. Colineau, J. Phalip, and A. Lampert, "The delivery of multimedia presentations in a graphical user interface environment," in *International Conference on Intelligent User Interfaces*. New York, New York, USA: ACM Press, 2006, pp. 279–282.

[24] C.-f. Shu, A. Hampapur, L. Brown, J. Connell, A. Senior, and T. YingLi, "IBM smart surveillance system (S3): a open and extensible framework for event based surveillance," in *International Conference on Advanced Video and Signal Based Surveillance*. IEEE, 2005, pp. 318–323.

[25] IBM, "IBM Smart Surveillance System." [Online]. Available: http://www.research.ibm.com/peoplevision/

[26] Siemens, "Siveillance Vantage," 2012. [Online]. Available: http://www.buildingtechnologies.siemens.com

[27] ——, "Siveillance SiteIQ Wide Area," 2012. [Online]. Available: http://www.buildingtechnologies.siemens.com

[28] Ipsotek, "Tag and Track," 2011. [Online]. Available: http://www.ipsotek.com/?q=en/news/48

[29] G. L. Foresti, C. Micheloni, and C. Piciarelli, "Detecting moving people in video streams," *Pattern Recognition Letters*, vol. 26, no. 14, pp. 2232–2243, Oct. 2005.

[30] C. Piciarelli and G. L. Foresti, "Online Trajectory Clustering for Anomalous Event Detection," *Pattern Recognition Letters*, vol. 27, pp. 1835–1842, 2006.

[31] C. Micheloni, B. Rinner, and G. Foresti, "Video Analysis in Pan-Tilt-Zoom Camera Networks," *IEEE Signal Process. Mag.*, vol. 27, no. 5, pp. 78–90, Sep. 2010.

[32] C. Piciarelli, C. Micheloni, and G. L. Foresti, "Occlusion-aware multiple camera reconfiguration," in *International Conference on Distributed Smart Cameras*. Atlanta, GA, USA: ACM Press, 2010, pp. 88–94.

[33] S. Burigat, L. Chittaro, and S. Gabrielli, "Navigation Techniques for Small-screen Devices: an Evaluation on Maps and Web pages," *International Journal of Human-Computer Studies*, vol. 66, no. 2, pp. 78–97, Feb. 2008.

[34] Ergonomics of human-system interaction – Part 210: Human-centred design for interactive systems, "ISO Standard 9241-210:2010," 2010.

[35] J. Nielsen and R. Molich, "Heuristic evaluation of user interfaces," in *SIGCHI Conference on Human Factors in Computing Systems Empowering People*, ser. CHI '90. New York, New York, USA: ACM Press, 1990, pp. 249–256.

[36] G. P. Polson, C. Lewis, J. Rieman, and C. Wharton, "Cognitive walkthroughs: a method for theory-based evaluation of user interfaces," *International Journal of Man-Machine Studies*, vol. 36, no. 5, pp. 741–773, May 1992.

[37] J. Nielsen, "Heuristic Evaluation," in *Usability Inspection Methods*, 1st ed., J. Nielsen and R. L. Mack, Eds. New York, New York, USA: John Wiley & Sons, 1994, p. 448.

[38] N. Martinel and C. Micheloni, "Re-identify people in wide area camera network," in *International Conference on Computer Vision and Pattern Recognition Workshops*. Providence, RI: IEEE, Jun. 2012, pp. 31–36.

Niki Martinel (B.Sc. '08, M.Sc. '10) received the Laurea degree (cum laude) in Multimedia Communications from the University of Udine, Udine, Italy. Since 2011 he is a Ph.D. Student in Multimedia Communication and he is a member of the AVIRES Lab in the Department of Mathematics and Computer Science at the same university. His research interests include wide area scene analysis, patter recognition techniques for surveillance applications, machine learning, feature transformations and Human-Computer Interaction. He is student member of IEEE and IAPR.

Christian Micheloni (M.Sc. '02, Ph.D. '06) received the Laurea degree (cum Laude) as well as a Ph.D. in Computer Science from the University of Udine, Udine, Italy.Since 2007, he is assistant professor in Computer Science. Since 2000 he has taken part to different national and European projects. He has co-authored different scientific works published in International Journals and Refereed International Conferences. He serves as a reviewer for several International Journals and Conferences. He operated as chairman and member of Technical Committees at several conferences. Dr. Micheloni's main interests involve active vision, neural networks, camera networks, self reconfiguring camera network. He is also interested in pattern recognition and machine learning. He is member of the IAPR and IEEE.

Claudio Piciarelli received the M.Sc. and Ph.D. degrees in Computer Science from University of Udine, Italy, in 2003 and 2008 respectively. Since 1999 he has been working with the Artificial Vision and Real-time Lab, University of Udine. He is an assistant professor at the University of Udine. Dr. Piciarelli authored or co-authored more than 30 works published in international journals and conferences and actively serves as a reviewer for several IEEE journal and conferences. His main research interests include computer vision, artificial intelligence, pattern recognition, and machine learning. He worked on several national and European projects on topics such as traffic monitoring, airports surveillance and counter-terrorism systems. He is a member of the IEEE and IAPR.

Gian Luca Foresti was born in Savona (Italy) in 1965. He received the Laurea degree cum laude in Electronic Engineering and the Ph.D. in Computer Science from University of Genoa, Italy, in 1990 and in 1994, respectively. Since 1998 he is Professor of Computer Science at the Department of Mathematics and Computer Science, University of Udine, and Director of the Artificial Vision and Real-Time System (AVIRES) Lab. Prof. Foresti is author or co-author of more than 120 papers published in International Journals and Refereed International Conferences. He was general co-chair, chairman and member of Technical Committees at several conferences. He has served as a reviewer for several international journals, and for the European Union in different research programs. He is Senior member of IEEE and Fellow member of IAPR.