

Pre-emptive Camera Activation for Video-Surveillance HCI

Niki Martinel, Christian Micheloni, and Claudio Piciarelli

Università degli Studi di Udine

Abstract. Video analytics has become a very important topic in computer vision. Many applications and different approaches have been proposed in different fields. This paper introduces a new information visualisation technique that aims to reduce the mental effort of security operators. A video analytics and a HCI module have been developed to reach the desired goal. Video analysis are exploited to compute possible trajectories used by the HCI module to pre-emptively activate cameras that will be probably interested by the motion of detected objects. The visualisation of most interesting views is provided to reduce the mental effort of human operators and allow them to follow the object of interest. Usability tests show the efficiency of the proposed solution.

1 Introduction

Video Surveillance Systems (VSS) are common in commercial, industrial, and now also in residential environment. The proliferation of VSSs and the increasing number of cameras per installation have brought new kind of problems. One of these is the usability of User Interfaces (UI).

Common video analytics tasks generally ask for significant mental efforts. Because of this, the limit of human attention is often overtaken and the real effectiveness, given the high number of video streams, tends to become a negative factor.

Today, this aspect is increasingly important because the primary goal of surveillance research is focused on the automatic understanding of the activities across wide areas [7]. When suspicious people have to be followed through multiple cameras users still experience great difficulties. For such a reason, VSSs must provide effective UIs such that the information represented by video streams and related data could be really understood by the end-users. User needs and application properties must be considered during the UIs development. Within last years many systems have been equipped with huge wall screens and/or some remote smaller displays. Anyhow, the required mental effort is not significantly decreased.

To follow objects between camera views users often employ a single monitor which dimensions are generally too small [3]. This kind of activity requires the user a huge mental effort and tends to become harder and harder if done for a long time. To sidestep such a problem, Iannizzotto et al. in [6] proposed a

perceptual user interface that allows users interaction by means of gestures. In [5], Girgensohn et al. developed a Dynamic Object Tracking Systems that provides geographical cues [13] about the monitored environment. Morris and Trivedi in [9] and Bottoni et al. in [1] described similar solutions. According to these works new VSSs still have to strongly reduce the required mental effort.

The development of an effective and powerful information visualisation technique is the main goal of this work. The novel aspect of the paper is represented by the selection of proper video streams, their organisation and alternation. For such purposes, objects, i.e. camera views, are chosen and activated with a significant difference: rather than displaying all available camera views, only most probable streams, i.e. those that will be interested by the object motion, are presented to the user. To determine most probable streams, the system must foresee the object trajectories and the cameras that best acquire such possible paths.

In [12], Qureshi and Terzopoulos describe how it is possible to activate different camera views in order to track a single object among different fields-of-view (FOV) that are geographically adjacent. Thus, merging gained knowledge about trajectory prediction and camera view selection and activation, the proposed work introduces a new way of showing visual information. The user interface developed is the result of an accurate process of camera view planning and selection. The main algorithm, according to space properties and a trajectory prediction tool, builds an activation plan to represent only those views that follow the predicted path of the object of interest.

Another novel aspect of this paper is related to the visualisation of geographical information about the monitored environment. Common desktop solutions, that made use of topographic maps, propose only a single topological representation of the environment. In addition, the proposed UI advantages from information visualisation studies conducted among mobile devices[2]. These exploit the introduction of the *detail plus overview* representation technique [14].

2 System Description

As shown in Fig.1 the architecture of the proposed VSS is organised in two main modules: a) a Video Analytics Module (VAM) and b) a Human-Computer Interface (HCI). The VAM processes the video streams generated by the cameras in order to analyse and identify events of interest [4] that should be provided to human operators together with useful information. For such a purpose a low level analysis module detects and recognises all the active objects in the environment. Then, the objects are tracked in order to provide temporal information about their activities. These are analysed by an event analysis module able to correlate the objects activities through time and space. Such an analysis is used by a trajectory estimator [11] that has the goal to foresee the trajectories of the objects of interest given their activities and past trajectories. Hence, this module path-plans the actions of interest such that the cameras can be opportunistically tasked or redirected in order to improve the analysis capabilities. This is

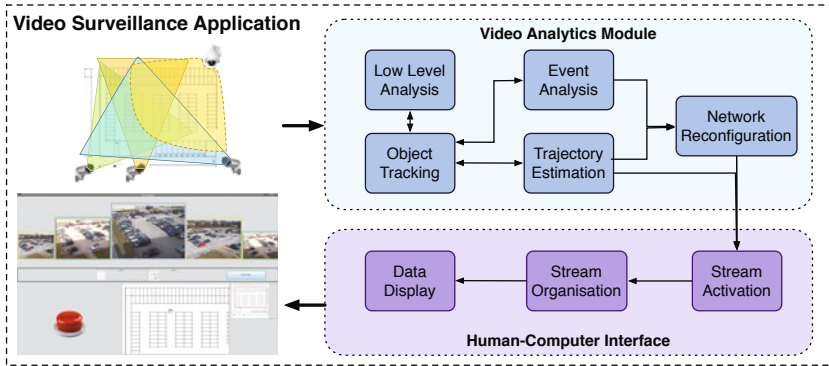


Fig. 1. Video Surveillance System

achieved by the network reconfiguration module [8]. The estimated trajectories and the new camera network configuration are given to the HCI module that has to optimise the cognitive capabilities of the operator. In this way, a first decision is taken about the most meaningful streams that have to be provided. This is done by considering the foreseen evolution of the environment. Once the most important streams have been decided a second module is in charge to determine how they must be organised on the user interface. Finally the streams are properly visualised on the UI together with useful information provided by the video analytics module.

3 Trajectory Analysis

The trajectory analysis module is fundamental to estimate the path that the object of interest most probably will take in the near future. This module is based on Piciarelli and Foresti work [11]. Let $Tr = \{(x_1, y_1), \dots, (x_n^t, y_n^t)\}$, where (x_j^t, y_j^t) is the position of the object expressed along the x^t and y^t axis of the map at time instant j , be the i -th trajectory detected by the trajectory extraction algorithm. The trajectories are grouped in clusters $C = \{(x_1^c, y_1^c, \sigma_1^2), \dots, (x_m^c, y_m^c, \sigma_m^2)\}$ where σ_j^2 is the local variance of the cluster at time j that statistically represent the most probable trajectories inside the monitored environment. In order to associate the current trajectory T to an available cluster C the following distance measure is adopted

$$D(T, C) = \frac{1}{n} \sum_{i=1}^n \min_j \left(\frac{\text{dist}((x_i^t, y_i^t), (x_j^c, y_j^c))}{\sigma_j} \right) \quad (1)$$

where dist is the Euclidean distance. Such a distance, representing the mean normalized distance of a trajectory point with the closest point of the cluster, is thresholded in order to associate it with a pre-computed cluster or to define a new one. Thus the matching process, as can be seen in Fig.2, allows to define a

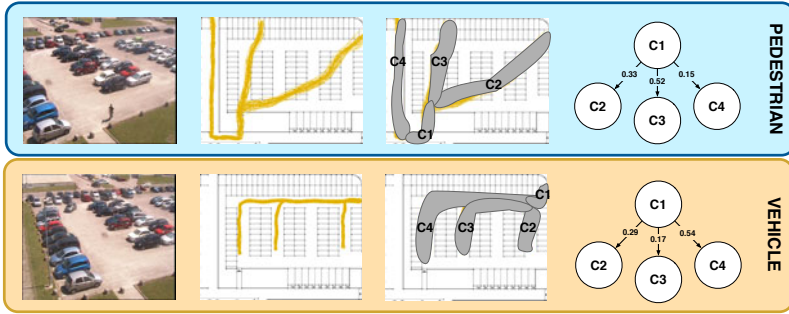


Fig. 2. Example of trajectory association and probability of future path

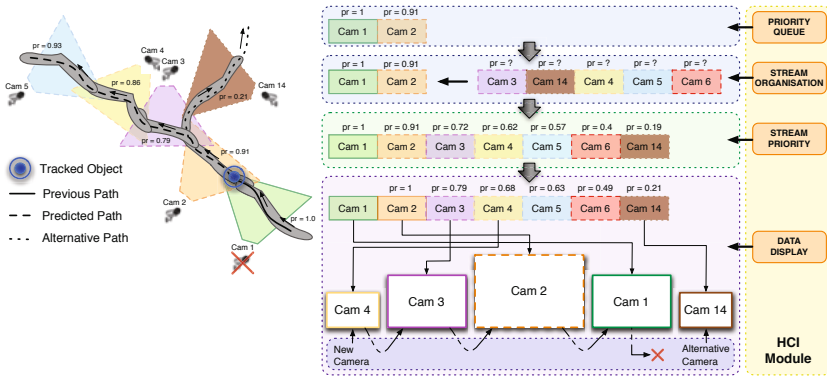


Fig. 3. HCI Module, example of stream organisation, activation and data display

probable path that the current object will follow. Matching the clusters positions with the cameras FOV allows to gain the probability that a sensor will acquire the object of interest in the near future. Thus, it is possible to have the activation probability of the cameras in order to correctly follow the object acquired by a camera.

4 HCI Module Definition

The HCI module is organised in three main components: a) the stream activation, b) the stream organisation and c) the data display.

4.1 Stream Activation

The stream activation makes active only those cameras that could be of interest for the operator. This HCI component analyses all available contents generated by the cameras and plans the hand-off between sensors [12] such that an object of interest could be continuously followed. The stream activation component provides an effective tool that is mainly based on the information provided by the

trajectory estimation and network reconfiguration modules. In particular, the trajectory analysis provides the estimated path of the object. This is correlated with the fields-of-view computed by the network reconfiguration module. In this way, the stream activation foresees the cameras that will, most probably, be interested by the motion of the object of interest. Thus, such cameras are included in a priority queue that will allow to continuously maintain the selected object on the interface.

Camera views are inserted into the priority queue as follows. If Q is the priority queue and cam_i is the i -th camera such that $i \in \{1, \dots, n\}$ where n represents the number of cameras, then cam_i is inserted in the priority queue if and only if

$$FOV_i \cap trajectory_j \neq \emptyset \quad (2)$$

where j is the j -th possible trajectory that the object could perform and FOV_i is the area on the ground covered by cam_i .

4.2 Stream Organisation

As show in Fig.3 this module has to re-weight all the streams that have been inserted in the priority queue. This is strictly necessary to allow a correct data interpretation. The goal is achieved by applying some organisation rules based on spatial relations between cameras views and object predicted trajectories. The algorithm is strongly dependent on the trajectories provided by the VAM module and by the stream activation component.

Streams that have been previously inserted into the priority queue are evaluated against the possible object trajectories by taking in account the geographical deployment of the sensors. Thus, according to the trajectory estimation component single view priority is calculated intersecting each trajectory cluster with each camera FOV. A priority value is then assigned to each camera.

The stream priority value is computed by traversing the predicted path tree (see Fig.2). Edges values $P(C_i, C_j)$, connecting the clusters C_i and C_j , represent the probability to move from cluster C_j to cluster C_i . Hence

$$P(C_i|C_j, C_{j-1} \dots, C_k) = P(C_i, C_j) \prod_{l=j}^{k+1} P(C_l, C_{l-1}) \quad (3)$$

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

4.3 Data Display

The described UI introduces a new way of displaying multiple video streams in a single device by using a strictly user-centred development process.

The introduced data display module has two main novelties that have been developed to improve the operator capabilities: a) the video streams area and b) the map area. The main UI component is represented by the video area that is based on the two previously described components. Such a UI element is able to display only those cameras that better support the operator activities necessary to monitor the selected object. In addition, to provide an effective user interface the map of the area is also displayed. On such a map, sensors and objects positions are displayed.

The data display module is object centred since the visualisation is independent of the number of cameras in the network. It depends only on the objects of interest tracked by the system. In this way, when tracking an object, the number of cameras interested by the task is fixed. If more objects of interest are present into the monitored area the operator is able to follow one of these just switching, through a tabbed pan, the active visualisation. The number of objects of interest that are associated to a single operator is limited.

Video Streams Area. The video streams area represents the most important UI component of this work. The visualisation techniques adopted have been tailored according to the results obtained from different evaluations conducted among a set of preselected users. As previously described video streams are arranged inside the priority queue following some basic rules linked to the spatial properties between cameras and the predicted trajectory of the tracked object (see section 4.2).

Following the priority queue, the most important view, i.e. the stream of the sensor whose FOV best shows the object (and has the highest priority), is always displayed at the centre of the streams area. The streams that came after the most important camera will be displayed at the side of the main camera view according to the movement of the object. The previous camera with the highest priority will be shown at the other side. So, thanks to this the operator can clearly see which is the previous and the next camera view that will be used to follow the object itself. Finally the camera view that has been assigned the highest priority and is not related to the main predicted path, i.e. the most probable alternative path, will be displayed at the extreme side of the main camera view such that if the tracked object does not follow the main predicted path the operator can still see it.

Assuming that the object is moving from right to left, the next view will be presented to the operator at the left of the current main view (see Fig.3). In this way, as the main view changes, the next predicted stream will be moved to the centre of the visualisation area. Adopting this technique the object moving out of the main camera view will appear in the next stream represented in the direction of the object itself.

The video stream area introduces another visualisation cue that aims to better support operator tasks. Such a cue is described by the size of a single video stream representation that has the goal to explain the relative importance of each camera view. Adopting the described view selection technique, together with the view size cue, elements importance could decay exponentially over time so that video

streams stay in view for some time after they become less important. In some cases, as the object move along its path, it could be possible that the camera selection and representation change too quickly inside the UI. This way, the user could get confused. So, to solve this possible problem the user interface introduces content animation. As long as the object follows the predicted path, camera views gradually slide in the opposite direction respect to the movement vector of the object. Old selected views are scaled down and animated out of the UI.

Other important aspects that have been introduced in the UI are represented by colours and by depiction techniques adopted to differentiate camera views. The UI introduces a colour-coded and a drawing style technique to discern camera views and to relate them to the represented objects in the map area. These UI features have been adopted to best fit all user needs and to reduce the operators mental effort. The main goal has been achieved using a color-coded technique (which complies with colour-blind people) and introducing a different representative style for the main active sensor. Hence, all users could clearly distinguish between environment sensors and in particular can immediately recognise main active camera.

Although providing only stream information could fit some operator requirements, in some cases it might be useful to have geographical information about the area and spatial cues about the identified objects that are followed by ambient sensors. Using a topological representation that cooperates with the previously described techniques is probably the best way that can lead to better usability results.

Map Area. In many VSSs personnel is often required to monitor many video streams and the introduction of a map component could help them and improve their ability to follow objects behaviour between different camera views. Usually, the map, that is integrated in this kind of user interfaces, displays the location of each camera, its FOV and the objects being tracked. Moreover the map can sometimes be panned and zoomed for retrieving more accurate details.

In this work three main novel aspect have been introduced on map visualisation. The first is represented by the colour-coded technique previously analysed (see section 4.3). The second is given by the use, for each camera, of a colour-coded shaded area representing the sensor FOV. Finally, taking care of all commonly adopted techniques, the work proposed through this paper also introduces a novel interaction paradigm that is usually employed in mobile devices applications: the *detail plus overview* technique. This powerful information visualisation technique significantly improve user ability to find and search objects inside the monitored environment. Adding a small representation of the topological area map allow operators to interact with this user interface element and easily navigate the environment maintaining a fixed view about the whole area. Furthermore, if the user has zoomed the map view to retrieve more accurate details about the tracked object path, and this is now outside the current view of the map, the user can continue to follow the object through the smaller representation. It could also pan the magnified view to update the focus on the new target position.

5 Experimental Results

As user-centred design principles pointed out, it’s not possible to evaluate a project just executing some test on a single prototype. Thus, the work described has been conceived and developed using an iterative process that has the goal to produce the best interaction between the operator and the machine. This process has led to the development of four different prototypes that have been evaluated using empirical and non-empirical methods.

First of all some basic information about classes of users, context of use and application scenarios have been identified. This initial step drives the process to the second evaluation stage that is composed by the retrieving process. This has to identify the using cases and the user basic knowledge necessary to execute these evaluations. Then, evaluations tests have been executed and results analysed to identify and fix every single interaction problem.

Empirical evaluations have been conducted among a set of about forty pre-identified end-users. They have been asked to execute six basic activities like viewing a past event, follow the tracked object and so on. Test sessions have then been executed in a controlled environment where the researcher maintains a detached behaviour and intervenes only when the end-user isn’t able to reach the requested result or has some question about the UI elements behaviour.

To get some quantitative evaluation of the designed and developed UI some indexes have been defined. The success rate index (*SR*) aims to show how much efficient is the user interface. It is given by:

$$SR = \frac{n_s}{n_t} \tag{4}$$

where n_s is the number of correct-end test and n_t is the total amount of conducted evaluations. The information that arise from the *SR* index can then be used to gain an overview about the efficiency and clearness of the UI.

To get a more clear and accurate point of view, on each UI, single results obtained from the evaluation process have been extracted. Such information has been taken in account to precisely found what are the problems related to single human-machine task.

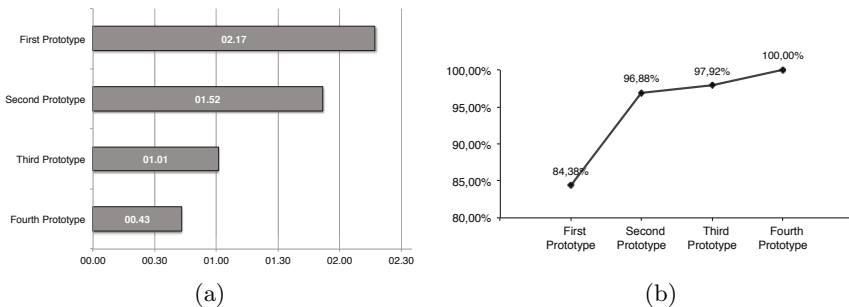


Fig. 4. Results obtained from experimental results. (a) Average Execution Time Index (b) Success Rate Index.

Similarly to the *SR* index the average execution time (*AET*) has been calculated to gain some information about the user interface efficiency. The *AET* is computed as

$$AET = \frac{\sum_{i=0}^{n_t} T_i}{n_t} \quad (5)$$

where T_i represents the i -th execution time needed to complete a single proposed task. This index has been used to represent how much time a single user needs to reach the given goal (user failure has been taken into account as well). In particular, data obtained from this index is as important as the data collected from the *SR* index because timing (especially in computer vision tasks) is much significant. So, analysing data it was possible to identify which were the HCI tasks that require a specific amount of time to be completed. During the design process, if a given task required too much time the UI elements involved in that process had to be reviewed.

Non-empirical evaluations have been executed with the direct support of four HCI experts that try to complete the predefined tasks and thus identify which problem an end-user could find. In particular during this kind of evaluation have been used two commonly adopted techniques: a) the heuristic evaluation [10] and b) the cognitive walkthrough [15]. In both cases results obtained from evaluations have been crossed with empirical evaluation data to get the best results as possible.

As shown in Fig.4 results obtained demonstrate the effectiveness of a user centred design research process that directly involve end-users in the evaluation task. The *AET* index that has been evaluated against each prototype shows a constant reduction of the time necessary to complete a requested task. It shows very interesting values and demonstrates the efficiency of the fourth prototype that average require less than a minute to complete a single task. The *SR* index describes the relevance of the adopted approach too and shows once again the clearness of the last evaluated prototype (i.e. 100% of successful tests).

6 Conclusions

In this paper a novel information visualisation approach for VSSs has been described. A VAM has been developed to identify and predict the path of object of interest. Tracking data is used to evaluate object trajectories and determine camera configurations. A HCI module has been used to select, organise and show best streams to keep the object inside the UI. Obtained results have shown that the adopted information visualisation technique is very efficient and leads to a mental effort reduction for end-users.

References

1. Bottoni, P., De Marsico, M., Leviardi, S., Ottieri, G., Pierro, M., Quaresima, D.: A dynamic environment for video surveillance. In: Gross, T., Gulliksen, J., Kotzé, P., Oestreicher, L., Palanque, P., Prates, R.O., Winckler, M. (eds.) INTERACT 2009, Part 2. LNCS, vol. 5727, pp. 892–895. Springer, Heidelberg (2009)

2. Chittaro, L.: Visualizing information on mobile devices. *IEEE Computer* 39(3), 34–39 (2006)
3. Colineau, N., Phalip, J., Lampert, A.: The delivery of multimedia presentations in a graphical user interface environment. In: 11th International Conference on Intelligent User Interface, Sydney, Australia, pp. 279–281 (February 2006)
4. Foresti, G., Micheloni, C., Piciarelli, C.: Detecting moving people in video streams. *Pattern Recognition Letters* 26, 2232–2243 (2005)
5. Girgensohn, A., Kimber, D., Vaughan, J., Yang, T., Shipman, F., Turner, T., Rieffel, E., Wilcox, L., Chen, F., Dunningan, T.: Dots: Support for effective video surveillance. In: Proceedings of the 15th International Conference on Multimedia, Augsburg, Germany, pp. 423–432 (September 2007)
6. Iannizzotto, G., Costanzo, C., Rosa, F.L., Lanzafame, P.: A multimodal perceptual user interface for video-surveillance environments. In: 7th International Conference on Multimodal Interfaces, pp. 45–52 (October 2005)
7. Lee, L., Romando, R., Stein, G.: Monitoring activities from multiple video streams: Establishing a common coordinate frame. *IEEE Transactions on Pattern Analysis And Machine Intelligence* 22(8), 758–767 (2000)
8. Micheloni, C., Rinner, B., Foresti, G.L.: Video analysis in pan-tilt-zoom camera networks. *IEEE Signal Processing Magazine* 27(5), 78–90 (2010)
9. Morris, B.T., Trivedi, M.M.: Contextual activity visualization from long-term video observations. *IEEE Intelligent Systems* 25(3), 50–62 (2010)
10. Nielsen, J.: Usability inspection methods. In: Conference Companion on Human Factors in Computing Systems, CHI 1994, pp. 413–414. ACM, New York (1994)
11. Piciarelli, C., Foresti, G.: Online trajectory clustering for anomalous event detection. *Pattern Recognition Letters* 27, 1835–1842 (2006)
12. Qureshi, F.Z., Terzopoulos, D.: Planning ahead for ptz camera assignment and handoff. In: Third ACM/IEEE International Conference on Distributed Smart Cameras 2009, Como, Italy, pp. 1–8 (August–September 2009)
13. Rieffel, E.G., Girgensohn, A., Kimber, D., Chen, T., Liu, Q.: Geometric tools for multicamera surveillance systems. In: First ACM/IEEE International Conference on Distributed Smart Cameras, pp. 132–139. ACM/IEEE (2007)
14. Spence, R.: Information Visualization. Addison Wesley, Harlow (2000)
15. Wharton, C., Rieman, J., Lewis, C., Polson, P.: Usability Inspection Methods, 1st edn., ch. 5, pp. 105–140. Wiley, Chichester (1994)