

Design of vision technology for automatic monitoring of unexpected events

Xavier Desurmont, Jerome Meessen, Arnaud Bastide, Christophe Parisot, Jean-Francois Delaigle
Video Analysis Group, Image Department, Multitel, Belgium

Abstract

Due to the emergence of digital standards and systems it is now possible to deploy easily and rapidly vision technology on site for permanent or temporary uses for automatic monitoring of unexpected events. Examples of challenging applications [1,2] are surveillance, traffic monitoring, marketing, etc. These techniques could be also useful for instrumental recording of human or animal behavior for research purpose. This talk describes a practical implementation of a distributed video system with emphasis on video hardware issues like acquisition and image processing necessary for useful event detection. The requirements for these systems are to be easy to use, robust and flexible. Our goals are to obtain efficiently implemented systems that can meet these strong industrial requirements. A computer cluster based approach with network connections is the innovative solution proposed. The main advantage of this approach is its flexibility. Since mobile objects are important in video surveillance, these systems will include image analysis tools such as segmentation and object tracking. First we present the typical requirements of such a system. We consider issues like the facility to deploy network-connected real-time multi-cameras, with reusable modular and generic technologies. Then we analyze how to cope with the needs to integrate a solution with state-of-the-art technologies. As an answer we then propose global system architecture and we describe its main features to explain each underlying module. To illustrate the applicability of the proposed system architecture in real case studies, we show some scenarios of deployment for indoors or outdoors applications.

Keywords

Multi-camera, real-time, computer vision, tracking, behavior.

1 Introduction

Video surveillance is a large market as the number of installed cameras can attest. Nevertheless, there is still a need for complete and generic systems that can be inserted in an existing camera network (e.g. CCTV) to increase intelligence and handle automatic processing. These technologies could also be used in traffic monitoring, marketing and general behavior recognition. The requirements for these systems are to be network-connected, multi-cameras, modular, the display must be user-friendly, the vision modules should be plug-and-play and the overall system must be highly reliable and robust. The work reported here has both research [3,4,5] and industrial motivations. In this article we present a generic, flexible and robust approach for an intelligent real-time video-analysis system.

The paper is organized as follows: section 2 describes the global system and its main characteristics; section 3 is devoted to the image analysis module. Section 4 gives some applications and section 5 concludes and indicates future work.

2 System overview

The video-analysis system presented in this paper [6] is based on digital network architecture. This kind of system can be deployed in a building, for instance or can be connected to an existing data network. Basically, the system is composed of computers connected together through a typical LAN. The various cameras are plugged either on an acquisition board on a PC or directly on the local network hub for IP cameras. A human computer interface and a storage space are also plugged on this system. The main advantage of such architecture is its flexibility. The logical architecture has been designed in a modular way to allow a fair resource allocation over the cluster. Future needs in computing power will be simply addressed by adding a PC in the cluster. Videos are compressed in MPEG4 video stream.

2.1 Hardware

Typical hardware could be smart network cameras (see figure 1 and 2) performing acquisition, processing (detection, video compression) and network communication. Usual network components like switches and hubs, storage units or other processing units and complete PC for user interaction and also illumination sub-systems for night vision are also part of the distributed system.

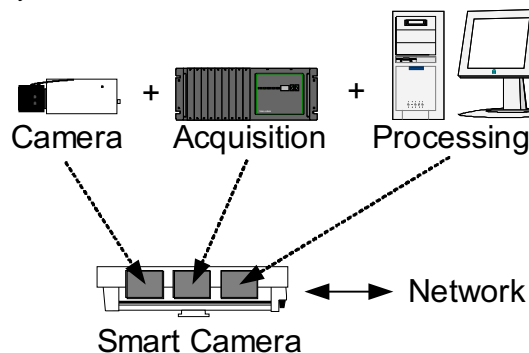


Figure 1. The smart network camera principle.

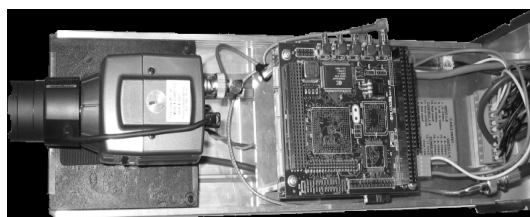


Figure 2. The ACIC SA smart camera CMVision (Photo by courtesy of ACIC SA).

3 Image analysis module

High-level interpretation of events within the scene requires low level vision computing of the image and of moving objects. It is usually needed to generate a representation for the appearance objects in the scene. For our system, the architecture of the vision part is divided in three main levels of computation that achieve the interpretation (figure 3):

- Image level (acquisition, image filtering, background evaluation and segmentation),
- Blob level (description, blobs filtering, matching, tracking description and filtering),
- Event level (tracking analysis, finite state machine, performance evaluations).

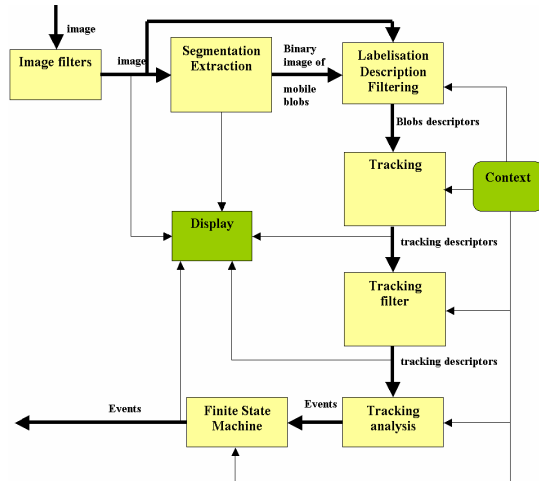


Figure 3. Design of the vision system components.

3.1 Segmentation

The common bottom-up approach for segmenting moving objects is the background estimation and foreground extraction (usually called background subtraction) [7]. Many reference image models could be used for representing the backgrounds as they are implemented in the system (low pass temporal recursive filter, median filter, unimodal Gaussian, mixture of Gaussians, vector quantization [8]). We are commonly using the mixture of Gaussians background, as it is quite robust to common noises such as monitor fluttering or branches moving in trees.

3.2 Blobs description and filtering

The aim of blobs description and filtering is to make the interface between foreground extraction and tracking and to simplify the information. The description process translates video data into a symbolic representation (i.e. descriptors). The goal is to reduce the amount of information to what is necessary for the tracking module. The description process calculates, from the image and the segmentation results at time t , the k different observed features of a blob i : 2D position in image, 3D position in the scene, bounding box, mean RGB color, 2D visual surface, inertial axis of blob shape, extreme points of the shape, probability to be a phantom blob, etc. At this point, there is another filtering process to remove small blobs, blobs in an area of the image not considered, etc. Other model-based vision descriptors could be also integrated for specific application such as vehicle or human 3D model parameters.

3.3 Tracking algorithm

As the other modules, the tracking part of the system is flexible and fully parametrical on-line. The set-up should be done for a trade-off between computational resources, needs of robustness and segmentation behavior. It is divided in four steps that follow a straightforward approach: estimation, cost matrix computation, matching decisions, tracks updates. Note that there are multiple predictions and cost matrixes when the last matching decision is not unique, and there are only multiple matching decisions for some matching algorithms in MHT (multiple hypothesis tracking [9]). Figure 4 briefly explains the architecture.

The tracking filtering is processed at the tracking description output. It is just as necessary as the other filters of the vision system. As usual the filter is used to remove the noise. At this level of processing, it can use the temporal consistency. We described above some types of filters that can be used in chain. Because the tracking description is a construction built piece by piece during the progression of the video sequence, it can process on-line or off-line. One filter detects and removes tracks that last for less that a fixed duration. This kind of noise comes when the segmentation detects noise in the image as an object. Another filter simplifies tracks by removing samples of blobs that give poor information (e.g. If the blob moves slowly). It could be seen as a dynamic re-sampling algorithm.

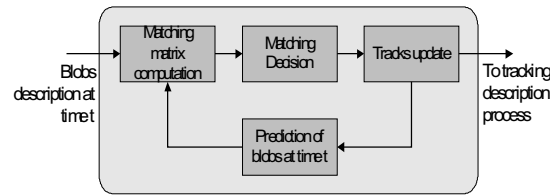


Figure 4. Basic architecture of Tracking.

3.4 Tracking description and filtering

The aim of tracking description and filtering is to make the interface between the tracking and the analysis processes and to simplify the information. The tracking description converts the internal tracking result to a graph (figure 5), and then it adds some useful information to the matching data. It computes the time of life of every blob of a track (i.e. the duration of the track from apparition to the specify blob), the time before death (i.e. the duration of the track to disappearance of the specify blob). It also describes a piece of track restricted in a small area as a stopped object. The grammar of the tracking description of blobs behavior includes apparition (new target), split, merge, disappearance, stopped, unattended object, entering a zone, exiting a zone.

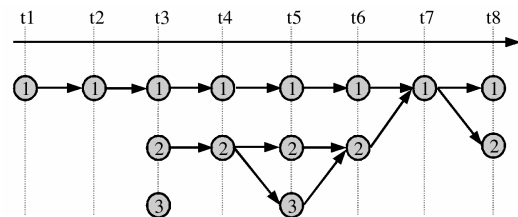


Figure 5. Internal tracking result description. $t1-t8$ are the time-stamps. Circles show objects in the image at time t and arrows show matchings between objects in different frames.

At the tracking description output, the tracking filtering is performed. It is as necessary as the other filters of the vision system. As usual the filter is used to remove the noise. At this level of processing, it can use temporal consistency. We describe below some types of filters that can be used in chain. Because the tracking description is a construction built piece by piece during the progression of the video sequence it can process on-line or off-line.

"*smalltrack*": It detects and removes tracks that last for less than a fixed duration, i.e. the delay between apparition and disappearance of the track. One can see on figure 6 the object labeled 3 at t3 has been erased by this filter.

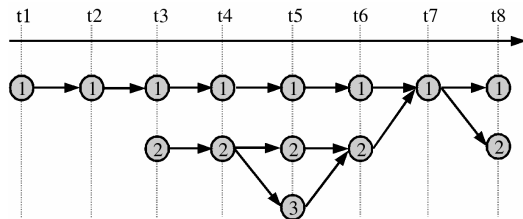


Figure 6. Output of "*smalltrack*" filter

"*simplifycurvetrack*": Simplifies tracks by removing samples of blobs that give poor information (i.e. if we delete it, we can interpolate it from other parts of the track). Figure 7 shows graphically the difference with and without this filter. Figure 8 shows the output in tracking description.

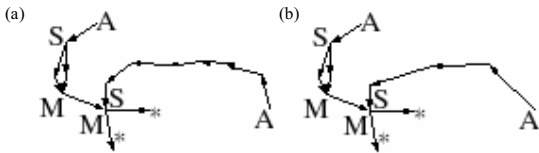


Figure 7. (a) raw tracking description, (b) tracking description filtered by "*simplifycurvetrack*". Symbols A, D, S, M and * mean respectively apparition, disappearance, split, merge and "object in current frame".

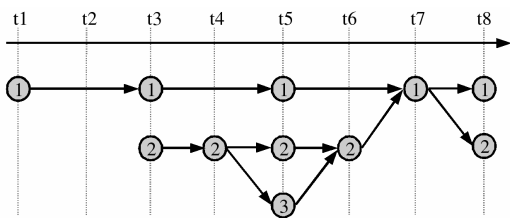


Figure 8. After filter "*simplifycurvetrack*" a track has been simplified by removing some objects instances.

"*simplifysplitmerge*": Removes one part of a double track stemming from a split and then a merge. This kind of noise comes when the segmentation detects two blobs when in all likelihood there is a unique object. Figure 9 and 10 shows the results.

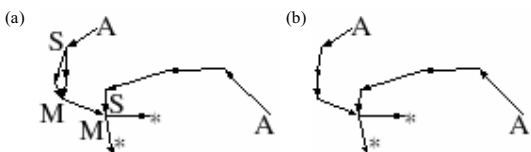


Figure 9. (a) raw tracking description, (b) tracking description filtered by "*simplifysplitmerge*"

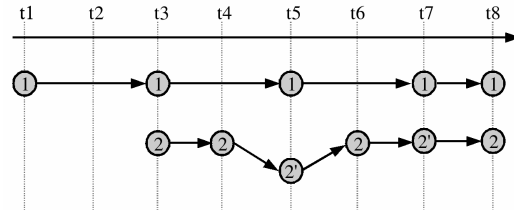


Figure 10. Output tracking after "*simplifysplitmerge*".

We do not describe here other implemented filters in detail. Another filter removes tracks that start or end in defined region of the image.

3.5 Tracking analysis and event generation

The tracking analysis is a process that receives the tracking description. It can find predefined patterns like objects entering in a defined zone of the image and exiting by another one, objects which have exceeded a certain speed limit, or immobile objects for a minimum time which stem from another mobile object. Figure 11 shows this particular pattern of tracking description. In this example we want to know how many people enter or not, with or without looking at the menu near the entrance door. The processing of this module is to look into the tracking description graph (e.g. figure 10) to find the predefined patterns.

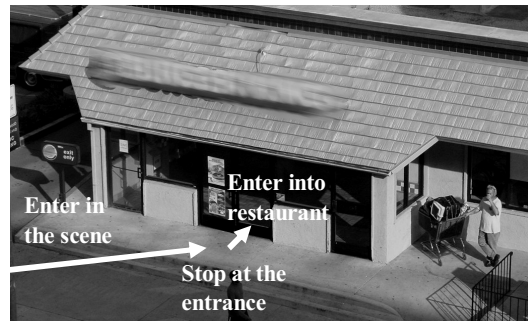


Figure 11. Tracking description pattern for "read menu before entering".

4 Applications

In this section, we give some applications of this kind of system. For vision systems, the representation of the scene is quite basic (positions of objects along the time). Thus it is difficult to automatically find people thoughts because, for example, of the difficulty to recover the facial expression. Therefore, to be automatically found, behavior should be easily visually understandable.

4.1 Traffic monitoring

These technologies are helpful to find automatically traffic jams, accidents, fires in tunnel, pedestrians near highways.



Figure 12. Example of detection of stopped car.

4.2 Surveillance:

There are many applications, like car park, public place (e.g. monitoring metro stations, detection of loitering, abandoned objects), private place, jails.



Figure 13. Example of car park surveillance scene.

4.3 Makerting

The purpose is to count people that enter and exit from each door of a shop. To know in which part and at what time people behave in the shop, according to some changes in goods layout.



Figure 14. View of a scene of a shopping center.

4.4 Behavior

The Relevance is to understand how humans or animals behave with automatic annotation of motion, movement and interactions. The interest of the system presented in this paper is that it is non intrusive, you don't need to put captors (like magnetic, inertial, GPS, etc.) or markers (like colors). After processing, intelligent content access is possible.



Figure 15. View of animals to find motion behavior.

5 Conclusion

In this paper, we have proposed an approach for video-analysis platform [6] that can provide the flexibility needed by researchers and that can meet the strong efficiency requirements of industrial applications. We have described the whole image analysis module with segmentation, tracking and event analysis. Originally it was dedicated to surveillance, but it could be used for others applications. These techniques could be useful for instrumental recording of human behavior. The basic limitation is the type of behavior: behavior should be visually understandable.

References:

1. A.Cavallaro, D. Douchamps, T. Ebrahimi and B. Macq, "Segmenting moving objects : the MODEST video object kernel", WIAMIS 2001, Workshop on Image Analysis for Multimedia Interactive Services, Tampere, Finland, May 16-17, 2001.
2. F. Cupillard, F. Brémont and M. Thonnat, "Tracking groups of people for video surveillance", 2nd European Workshop on AVBS Systems.
3. T. Shcoepflin, C. Lau, R. Garg, D. Kim and Y. Kim, "A research Environment for Developing and Testing Objet Tracking Algorithms", Proceedings of the SPIE, Electronic Imaging 2001, vol. 4310, pp. 667-675.
4. C. Jaynes, S. Webb, R. Steele and Q. Xiong, "An open development environment for evaluation of video surveillance systems", 3rd Int. Workshop on PETS, 1, 32-39
5. M. Valera and S.A. Velastin: "An Approach for Designing a Real-Time Intelligent Distributed Surveillance System", First Symposium on Intelligent Distributed Surveillance Systems (IDSS), IEE, 26 February 2003, London, pp.6/1-6/5
6. X. Desurmont, A. Bastide, C. Chaudy, C. Parisot, J.F. Delaigle and B. Macq, "Image Analysis Architectures and Techniques for Intelligent Surveillance Systems", Special Issue on Intelligent Distributed Surveillance Systems on the IEE Proc.-Vis. Image & Signal Process., Vol. 152, No. 2, April 2005.
7. A. Elgammal, R. Duraiswami, D. Harwood and L.S. Davis, «Background and Foreground Modeling Using Nonparametric Kernel Density Estimation for Visual Surveillance», Proceedings of the IEEE, vol.90, No. 7, July 2002
8. K. Kim, T. Horprasert, D. Harwood, L. Davis, "Codebook-based Background Subtraction and Performance Evaluation Methodology".
9. I.J. Cox and S.L. Hingorani, "An Efficient Implementation of Reid's Multiple Hypothesis Tracking Algorithm and Its Evaluation for the Purpose of Visual Tracking".

