

A SEAMLESS MODULAR IMAGE ANALYSIS ARCHITECTURE FOR SURVEILLANCE SYSTEMS

X. Desurmont^a, C. Chaudy^a, A. Bastide^a, J.F. Delaigle^a and B. Macq^b

{desurmont, delaigle}@multitel.be

^a, Multitel A.S.B.L, Belgium

^bUniversité catholique de Louvain, Belgium

Abstract

Video security is becoming more and more important today, as the number of installed cameras can attest. There are many challenging commercial applications to monitor people or vehicle traffic. The work reported here has both research and commercial motivations. Our goals are first to obtain an efficient intelligent system that can meet strong industrial surveillance system requirements and therefore be real-time, distributed, generic and robust. Our second goal is to have a development platform that allows researchers to imagine and easily test new vision algorithms thanks to its modularity and easy set-up. Previous papers [4,7] dealt with the core architecture for handling such problems as heterogeneous inputs, encoding, distribution, and storage. This paper will focus more precisely on the image analysis modules. We will consider the different kind of inputs, algorithm models as well as optimisation of memory, delay and genericity needs.

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. Examples of challenging applications [1] are monitoring metro stations [2] or detecting highways traffic jams, intelligent content access, detection of loitering. 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 [12,8,13] and industrial motivations. In this article we present a generic, flexible and robust approach for an intelligent real-time video-surveillance 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 concludes and indicates future work.

2. System overview

The CCTV system presented in this paper is based on a 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.

2.1 Data management

The system basic structure for data management between modules handles the concurrent access for sharing information for multiple producers and consumers. It optimises the memory needs by avoiding copies of data, the network communication (TCP/IP) for multiple instances of the data (compression handling if necessary), the dynamic connection and access to the stream, the buffer to absorb peak of processing, the implicit conversion of data (e.g. images from RGB to YUV), the propagation of consumers needs to producers (e.g. if no module uses a segmentation result, the segmentation module will be advised not to produce it), the monitoring of streams (performance, rate), the dispatching priority of data towards specific modules (it helps to process a real-time framework by decreasing latency). Fig.1 represents an example of a stream of data (control, source image, dynamic descriptors of scene, events) and a typical processing. The image acquisition is performed in each frame (25 fps), but tracking is done half-time (and therefore the interpretation too).

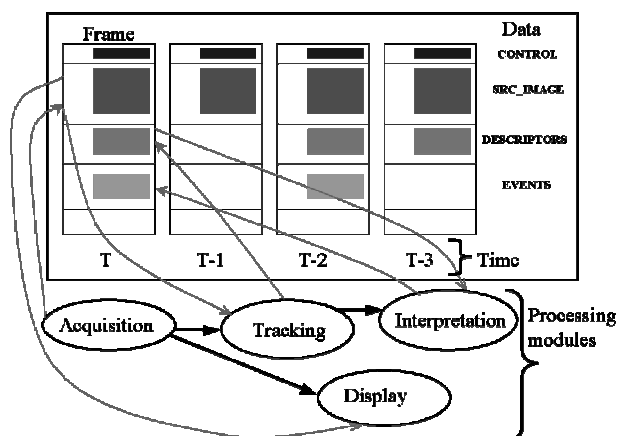


Fig. 1: Example of data management.

2.3 Asynchronous data handling.

Video Surveillance applications need to be real-time, because of the security aspect of minimum time reactions. Therefore video-surveillance requires low delay and timing constraints for processing. Classical real-time systems usually use internal periodical loop (E.g. a new image is acquired every 40 ms and should be processed, thus the process should last less than 40 ms). If the process costs more than 40 ms for one of the stages of a pipeline (if there is no pipeline, we consider the system as a unique-stage pipeline), a delay is introduced and grows for every new frame, this delay is limited to a maximum value (e.g. 200 ms), if it exceeds this limit, the system fails (e.g. the memory buffer is full, some data is lost and processing of this data, such as integration or derivation, will fail).

We overstep these kinds of problems by adding to each observation (image) a time-stamp, and then consider it to make all processing. (E.g. background adaptation, objects speed computing, etc.). Thus, if processing resources exceed the available system resources, some information is ignored but without disturbing the integrity of the system.

3. Image analysis module

High-level interpretation of events within the scene requires low level vision computing of the image and of the 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 (Fig. 2): 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). We will only concentrate on the description of the image level (Fig. 3).

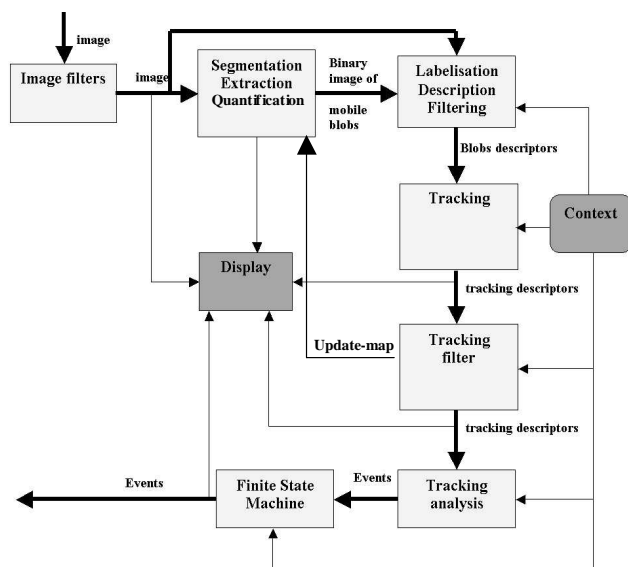


Fig.2: Design of the vision system components.

3.1 Acquisition and video file playing.

We are currently able to handle several protocols for image acquisition: IP (JPEG and MJPEG), IEEE1394 (raw and DV), wireless (analogic, Wifi) and composite (PAL, NTSC, SECAM). A time-stamp is attached to each frame at grabbing time, as this information is useful in the subsequent processing stages. For file replay, we are using the Ffmpeg [15] library that handles many codecs.

3.2 Calibration and 2D and 3D context

For some reasons it could be interesting to have a calibration of the camera (e.g. for multiple cameras application or when doing 3D with ground plane area). An easy and manual tool for calibration (Fig. 3) has been developed with the same procedure as in [14]. We also handle radial deformation via the Opencv [16] library. For fixed camera, a 2D context could be defined by users to identify areas in the image like input/output region, zone to ignore, etc. A 3D context (Fig. 4) could be set up to describe scene in 3D (the floor, the walls and the objects present at beginning of sequence).

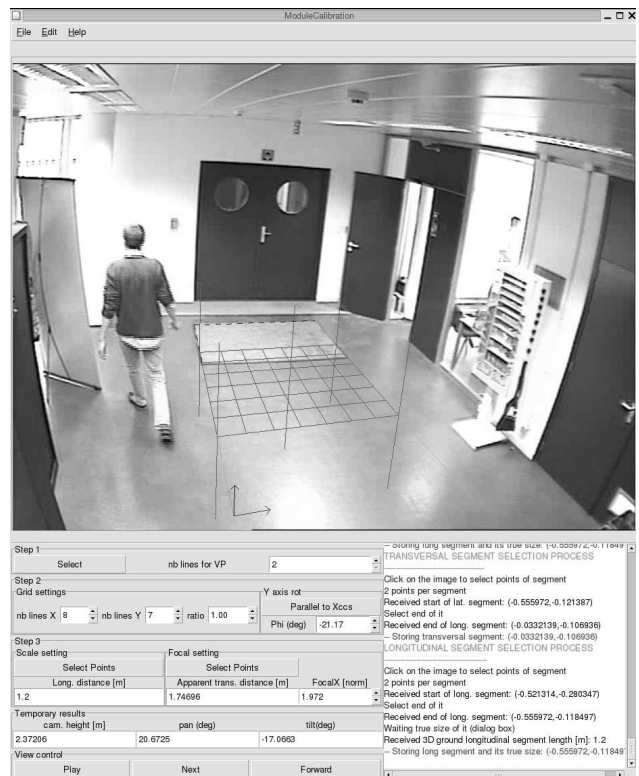


Fig.3: Interactive tool for manual calibration.

3.3 Image filtering

After acquisition, the current image is filtered in the pixel domain to decrease the spatial high-frequency noise. Usually the image is convoluted via a linear filter with a Gaussian kernel. But, in our scheme, we are using the optimised 2 passes exponential filter for less processing time. The image

is then downsized to reduce the need in computational resources of the segmentation process.

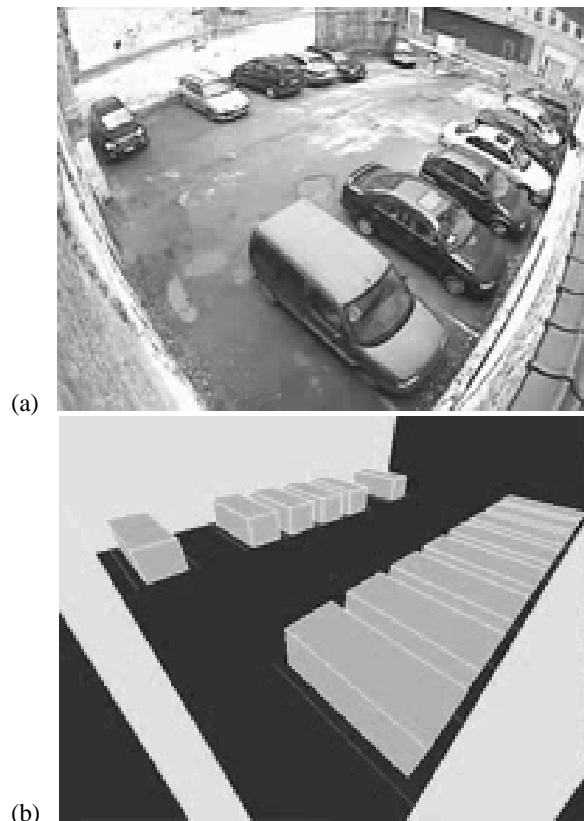


Fig.4: (a) Car park initial view from fixed camera, (b) Initial 3D context of the car park.

3.4 Automatic recalibration.

In the case of moving camera due to slight vibration (E.g. wind) or pan-tilt-zoom the image is altered and should undergo a 2D homographic geometric transformation [11] to be integrated back into the global 2D scene view (fig.5). At the current stage of development, the model is only handling pure translation (restricted area of pan-tilt with small view-angle).

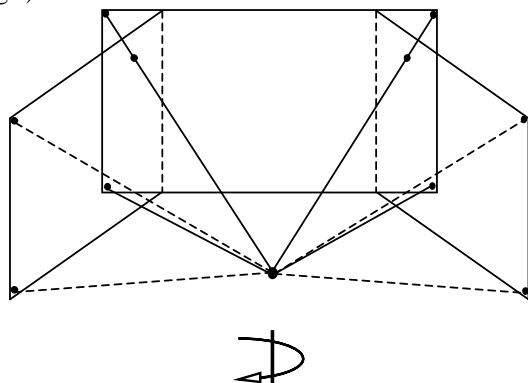


Fig.5: images from left and right view are projected into the middle image plan (scene view).

3.5 Segmentation

The common bottom-up approach for segmenting moving objects is the background estimation and foreground extraction (usually called background subtraction) [5]. 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 quantisation [9]). We are commonly using the mixture of Gaussians background, as it is quite robust to common noises such as monitor flatter or branches moving in trees. Some non-background segmentations like frame differencing are also implemented in the same design. However, these algorithms should satisfy the non-periodic processing framework of the system as described in 2.2. The segmentation is fully configurable as one can choose between all types of background models and parameters on-line (during the running of the system). Let us define $t_1 < t_2$ two times of following image acquisition. The process is divided in several parts (Fig. 6,7):

- 1) An *a priori* belief (probability) map is computed from the current image acquired at time t_2 and the background model updated at time t_1 (all pixels are considered independently). The most probable foreground is computed in that way: if the probability is superior to 0.5, the pixel is *a priori* (without knowledge of neighbourhood) considered as foreground.
- 2) If the pixel is inside a closed outline of pixel with a probability > 0.5 , the neighbourhood rule will consider it as foreground pixel. A second neighbourhood rule is applied: a pixel cannot be foreground if there is no path of connected *a priori* foreground pixel until a pixel of probability > 0.8 . These two rules permit hysteresis phenomena to decrease noise in the foreground.
- 3) Then two steps of decision are made at two different stages of the process to filter foreground objects: after the description (ignore zone, object too small, phantom object, etc...) and after the tracking (integrated object, etc.). We don't consider here these process (see subsection 3.6 and 3.7). After them, some foreground objects from 2) are discarded or some non-detected objects are added to the structure of data that contains foreground map. At this time, it is called update-map.
- 4) The background model is then updated with image at time t_2 for regions of the scene that the update map defines as background.

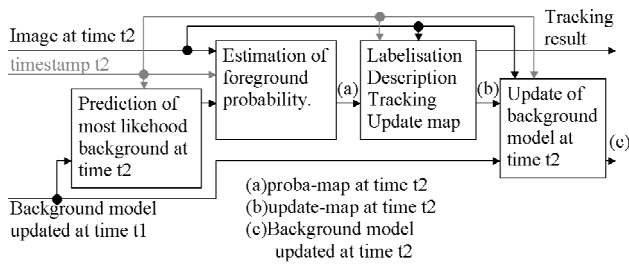


Fig.6: architecture of the segmentation process within the whole architecture.

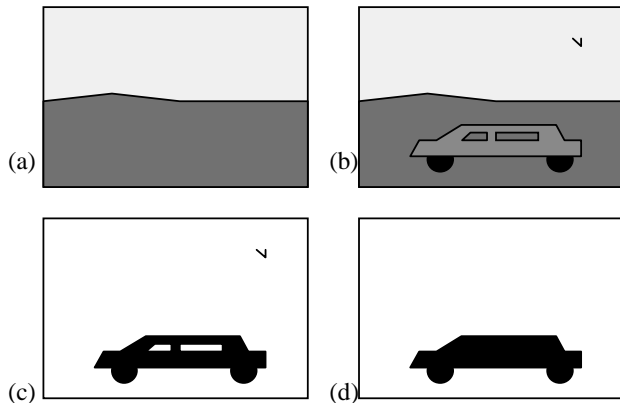


Fig.7 : (a) representation of background model at time t1 (b) image from the scene at time t2, (c) foreground at step 2, (d) foreground at step 3 (small blobs removed and holes also)

In Subsection 3.6 and 3.7, we will briefly investigate the processes that occurred between estimation 1) and 4).

3.6 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 colour, 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.

3.7 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 behaviour. 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 in MHT (multiple hypothesis tracking [3]). Fig. 7 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 than 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 resampling algorithm.

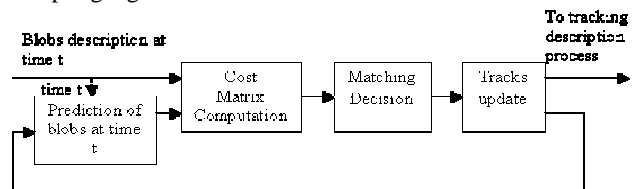


Fig. 8. Basic architecture of Tracking.

4 Conclusion

In this paper, we have proposed an approach for a third-generation video-surveillance platform [10] that can provide the flexibility needed by researchers and that can meet the strong efficiency requirements of industrial applications. This system is presented as being very efficient and it has been validated on various uses.

We described the whole image analysis module with a deeper description of the segmentation and description process.

We are currently investigating new vision modules, e.g. better segmentation and tracking methods. Moreover, other extensions and improvements will be made on the global system. Future works will also integrate full evaluation of best parameters and methods by comparing and analyse [6] the output of the global system for a specific application and the groundtruth.

Acknowledgements

This work has been granted by the Walloon Region under the FEDER project 171, the FIRST SPIN OFF program.

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. Brémond and M. Thonnat, "Tracking groups of people for video surveillance", 2nd European Workshop on AVBS Systems.
- [3] 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".
- [4] X. Desurmont J.F. Delaigle, A. Bastide, B. Macq "A generic flexible and robust approach for intelligent real-time video-surveillance systems", Manuscript of the oral presentation for the Real-Time VIII, 22 January 2004, Proceedings of SPIE Vol. 5297.
- [5] 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
- [6] T. Ellis, "Performance Metrics and Methods for Tracking in Surveillance", Information Engineering Centre, School of Engineering, City University, London. Proceeding 3rd IEEE Int. Workshop on PETS, Copenhagen, June 1 2002.
- [7] B. Georis, X. Desurmont, D. Demaret, J.F. Delaigle and B. Macq, "IP-Distributed computer-aided video-surveillance system", *Intelligent Distributed Surveillance Systems Workshop*, London, UK, February 26, 2003.
- [8] 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
- [9] K. Kim, T. Horprasert, D. Harwood, L. Davis, "Codebook-based Background Subtraction and Performance Evaluation Methodology",
- [10] L. Marcenara, F. Oberti, L. Foresti and C. Regazzoni , "Distributed architectures and logical-task decomposition in multimedia surveillance systems", Video Communications Processing and Understanding for 3GSS, Proceedings of the IEEE, 89, 1419-1440.
- [11] K. Okuma "Automatic Acquisition of Motion Trajectories: Tracking Hockey Players", M.Sc. Thesis, the University of British Columbia, May 2003.
- [12] T. Shcoepflin, C. Lau, R. Garg, D. Kim and Y. Kim, "A research Environment for Developing and Testing Object Tracking Algorithms", Proceedings of the SPIE, Electronic Imaging 2001, vol. 4310, pp. 667-675.
- [13] 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.
- [14] A D Worrall, G D Sullivan and K D Baker, "A simple, intuitive camera calibration tool for natural images", Department of Computer Science, The University of Reading, Berkshire, RG6 2AY, UK.
- [15] <http://ffmpeg.sourceforge.net/>
- [16] <http://sourceforge.net/projects/opencvlibrary/>