"A Seamless Modular Approach for Real-Time Video Analysis for Surveillance"

desurmont, Xavier ; Bastide, Arnaud ; Delaigle, Jean-Francois ; Macq, Benoît

ABSTRACT

In this paper we present a generic, flexible and robust approach for an intelligent real-time visual surveillance system. The proposed architecture integrates advanced video analysis tools, such as segmentation, tracking and events detection. The goal of these advanced tools is to provide help to operators by detecting events of interest in visual scenes. The approach is new because it is a seamless collaboration of many processing techniques from image space to dynamic scene space. We describe thoroughly an important section the vision system and prove its intrinsic modularity. The system is then demonstrated in an antiterrorist scenario of automatic detection of events in public facilities. A performance evaluation of the system is performed.

CITE THIS VERSION

A SEAMLESS MODULAR APPROACH FOR REAL-TIME VIDEO ANALYSIS FOR SURVEILLANCE

Xavier Desurmont[a], Arnaud Bastide[a] , Jean-François Delaigle[a], Benoît Macq[b]

[a] {desurmont, bastide, delaigle}@multitel.be, Multitel A.S.B.L, Mons, Belgium.
[b] macq@tele.ucl.ac.be, Communications and Remote Sensing Laboratory,
Université catholique de Louvain, Louvain-la-neuve, Belgium.

ABSTRACT

In this paper we present a generic, flexible and robust approach for an intelligent real-time visual surveillance system. The proposed architecture integrates advanced video analysis tools, such as segmentation, tracking and events detection. The goal of these advanced tools is to provide help to operators by detecting events of interest in visual scenes. The approach is new because it is a seamless collaboration of many processing techniques from image space to dynamic scene space. We describe thoroughly an important section the vision system and prove its intrinsic modularity. The system is then demonstrated in an antiterrorist scenario of automatic detection of events in public facilities. A performance evaluation of the system is performed.

1. INTRODUCTION

Video security is becoming more and more important today, as the number of installed cameras can attest. Examples of challenging applications [1] are monitoring metro stations [2] or detecting highways traffic jam, detection of loitering. In public facilities such as airports, there is a risk of terrorist attack by abandoning a bag or an object containing a bomb. The interest of detecting without delay such a situation might permit to avoid an explosion. In this paper we present a visual surveillance system that is set up for automatic detection of such unattended objects.

The work reported here has both research and commercial motivations. Our goals are first to obtain an efficient system that can meet the strong CCTV industrial requirements and second to have a development platform that allows researchers to imagine and test new vision algorithms. Such a system must for example include evaluation facilities, such as [3]. We report here an evaluation of the system with a wide range of sequences.

The paper is organized as follows: section 2 describes the global system and its main characteristics; section 3 goes deeper in the understanding of the vision system. Section 4 is devoted to performance evaluation of the system and section 5 concludes and indicates future work.

2. OVERALL SYSTEM OVERVIEW

The global architecture system [4] is composed of heterogeneous computers and cameras connected together through a network. A human computer interface and a storage space are also plugged onto this system. An easy-to-use and manual tool for camera calibration and other contextual data has been developed. In order to test and benchmark our algorithm on recorded particular sequences, a general video streams player was added.

3. VISION SYSTEM DESCRIPTION

The architecture of the vision part of the system is divided in three main levels of computation that achieve the interpretation (cf. Figure 1):

- Image level (image filtering, background evaluation and segmentation): Section 3.2
- Blob level (description, blobs filtering, matching, tracking description and filtering): Sections 3.3, 3.4 and 3.5
- Event level (tracking analysis, finite state machine, performance evaluations): Section 3.6

3.2. Pre-processing and segmentation

After acquisition, the current image is filtered in the pixel domain to decrease the spatial high frequency noise (convolution with a Gaussian kernel). The image is then downsized to reduce the need in computational resource of the segmentation process.

Many reference image models could be used for representing the background as they are implemented in the system. However, the segmentation used in the present application is based on the model developed in [5]. Basically the background is modeled as a mixture of Gaussians for each pixel. It is quite robust to common noises as monitor flattering or branches moving in trees.

3.3. Blobs description and filtering

The aim of blobs description and filtering is to make the interface between segmentation and tracking and simplify the information. The description process translates video data into a symbolic representation (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 $f_{k,i}^t$ 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, 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 as vehicle or human 3D model parameters.
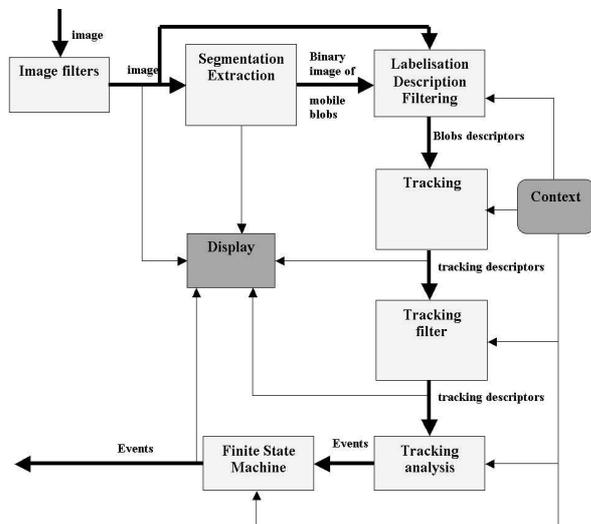
Figure 1: Design of the vision system components.

3.4. Tracking algorithm

The tracking part of the system is flexible and fully parametrical. The set up should be done for a trade off between computational resources and needs of robustness. It is divided in four steps that follow a straightforward approach:

- Estimator,
- Cost matrix,
- Matching decision(s),
- Tracks update(s).

For some matching algorithms (E.g. Multiple Hypothesis Tracking [6]) there are multiple matching decisions. Thus, there are multiple predictions and cost matrixes if the last matching decision was not unique. The Figure 2 explains briefly the architecture of the tracking.

-Estimator: The estimator predicts the blob's features (position, color, size, etc.). It is integrated as a recursive estimator to handle various estimator cores like Kalman filter [7], explicit Euler, etc. The estimation is then taken as the maximum a posteriori probability (MAP).

$$\hat{f}_{k,i}^{t} = \arg\max_{f_{k,i}^{t}} p(f_{k,i}^{t} | f_{k,i}^{1:t-1}) \qquad (1)$$

-Cost matrix computation: The aim of this matrix is to identify a cost $C_{i,j}$ for a matching between blobs i in current frame and j in current precedent frame (or the estimation of it in current frame). The cost is performed by a multi-dimensional data fusion Φ (2) that could be simple as Euclidian norm to complex like Dempster-Shafer fusion. Typically this cost is low when blobs look similar and high otherwise (visual features). It is also low when blobs are consistent in term of movement along the scene (3D positions).

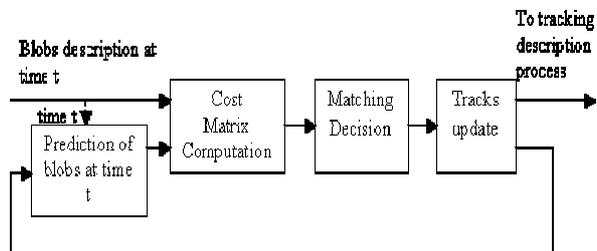$$C_{i,j} = \Phi_{k=1...N} d_k(f_{k,i}^{t}, \hat{f}_{k,j}^{t}) \qquad (2)$$

Figure 2: Basic architecture of Tracking.

-Matching: This part deals with the choice of several matching matrixes (or only one for unique hypothesis tracking). A matching matrix M is a Boolean matrix where $M_{i,j}$ is false when there is no link between blob i in current frame and blob j in precedent frame and true when there is a link. (E.g. if $M_{i,j}$ and $M_{k,j}$ are true, there are two links, thus the blob j in precedent frame splits in two blobs i and k in current frame). From a given cost matrix, there are many potential hypothesis matrixes for the real matching matrix [6]. Three algorithms can be executed: FMHM (full multi-hypothesis matching) and LMHM (light multi-hypothesis matching) and VSM (very simple matching). Note that for each iteration, one can permute the matching algorithm by another.

- FMHM: all hypothesis of matching are tested, the K best global hypothesis are kept. It takes a lot of computation time. E.g.: from the K best global previous hypothesis in a case of 6 blobs per frame, the number of hypothesis to test is $K \times 2^{6x6}$.
- LMHT: It reduces the number of matching hypothesis matrix. For a given blob in current image, it only performs matching hypothesis between the N blobs in precedent image witch have the lowest costs.
- VSM: This should be the simplest matching algorithm. Every blob from the precedent frame is matched to it nearest blob (lowest cost) in the current frame. Furthermore every blob from the current frame is matched to his nearest blob in the precedent frame. However if the cost between two blobs exceeds a threshold, the match is removed.

3.5. Tracking description and filtering

The aim of the tracking description and filtering is to present the tracking info to the tracking analysis. It simplifies and complements the output of the core tracking (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 dead (i.e. the duration of the track to disappearance of the specify blob). It also describes a segment of track restricted in a small area as a stopped object.

At the tracking description output, the tracking filtering is processed. It is as much 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. "Smalltrack" 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 blob. "Simplifycurvetrack" simplifies tracks by removing samples of blobs that gives poor information, (E.g. If the blob moves slowly. It could be seen as a dynamic re-sampling algorithm). We don't describe deeply other filters implemented.
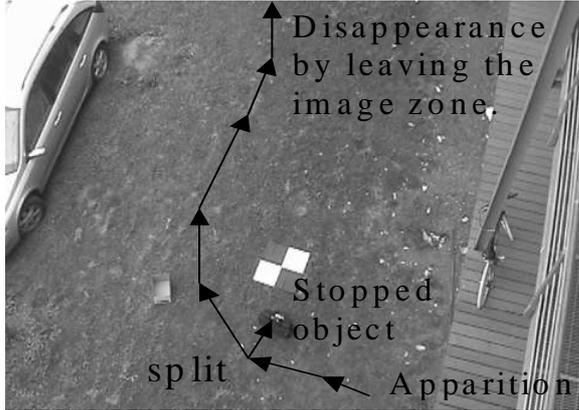

Figure 3: tracking description pattern for event "unattended object".

System Observation	Ground truth	
	Positive	Negative
Positive	Ntp (true positives)	Nfp (false positives)
Negative	Nfn (false negatives)	Ntn (true negatives)

Table 1: Boolean contingency table. Ntp is the number of observations confirmed by the ground truth. Nfp is the number of observations not matched in the ground truth. Nfn is the number of observations erroneously accepted as belonging to the ground truth. Ntn is the number of observations rejected as belonging to the ground truth.

Name	Expression
Detection rate (sensitivity)	Ntp/(Ntp+Nfn)
False positive rate	Nfp/(Nfp+Ntn)

Table 2: basic values for benchmarking.

3.6. Tracking analysis and event generation

The tracking analysis is a process that receives the filtered tracking description. It can find predefined patterns like objects entering from a defined zone of the image and exiting by another one, or objects witch have exceeded a certain limit speed, or also stopped objects for a minimum time which stems from another mobile object, Figure 3 shows this particular pattern of tracking description. In our application we are using this last pattern recognition as unattended object. Thanks to the tracking filtering, the tracking analysis is quite easy to perform. A similar approach has been described in [7].

To summarize, the ontology of tracking description of blobs behavior includes apparition (new target), split, merge, disappearance, stopped, unattended object, entering a zone, exiting a zone and occlusion. For apparition, there is no "confirm target" as in [7] because the filter "smalltrack" remove tracks that last for less than a fixed duration.

Table 3: description of ground truth of sequences.

4. RESULTS AND PERFORMANCE

We report hereunder the results obtained with the case study, which is the surveillance of areas from unattended object deposits. The goal of a computer vision system is to answer a particular problem with high robustness. Accordingly, we decided to test it on 217 sequences taken from 7 sets of sequences with a cumulated length of almost two hours of video with activities at almost all time. We subjectively consider that this is equivalent to 2 days of normal activities in a given scene.

For each sequence, the system should give a Boolean value for unattended object or not. We will compare these two populations of values (positive and negative) from observations (ground truth) against responses from the system. We use the metric described in [8] to characterize the success and failures of the algorithm. Tables 1 and 2 remind the scoring process.

4.1. Parameter setting for processing

For the experimentations described beside, the parameters used for the set up were set to the following values: the number of paths kept at each iteration was set to 1 (So the tracking is not

set to multi-hypothesis). The matching heuristic is VSM. The tracking filters delete phantom objects (virtual objects that occur when background objects start moving), small tracks. The event to detect is "unattended Object for 5 seconds".

These values were set up manually by expert decisions. In future, these could be learned directly by using ground truth sequences.

4.2. Sequences.

All sequences are color 25 fps and were taken from fixed PAL cameras. As a result of the medium content complexity and the segmentation quality requirement [9], the sequences were seen at 5fps sequences and the images were downsized to 192x144 pixels during the experimentation. Table 4 describes the ground truth.

4.3. Results and discussion

The table 5 shows a detection rate of 100% but a false positive rate of 18%. This means that 18% of the negative sequences trigger a false alarm, but some sequences lasts more than 5 minutes. If we consider that the 217 sequences represent 2 days of activities, thus there is less than one false alarm per hour, and none of the good detections are dismissed. The first hard requirement for a security system is to achieve a detection rate of nearly 100%. Nevertheless one of the most frustrating problems for users of such systems is the false activation of alarms [10]. That's why it is usually required to have a few false alarms. However, visual surveillance permits to check the image of the scene directly in order to break the alarm.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed an original approach for an automatic video surveillance platform that can provide the flexibility needed by researchers. We focus our attention to the vision system and demonstrate its architecture on automatic unattended object detection.

Future work will aim at lowering the false positive rate by overcoming different aspects of the system as segmentation, tracking but also by improving efficiency by auto-evaluation of the parameters set up. We are now implementing the codebook-based segmentation [11] and a tracking inspired by clustered particle filter [12].

ACKNOWLEDGEMENTS

REFERENCES

[1] Andrea Cavallaro, Damien Douxchamps, Touradj Ebrahimi & Benoit 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] Cupillard F, Brémond F and Thonnat M, "Tracking groups of people for video surveillance", 2nd European Workshop on AVBS Systems.

Number of sequences	Ground truth	
	Positive	Negative
217	26	191

Table 4: description of ground truth of sequences.

Nb sequen-ces	Ground truth		System observation			
	Posi-tive	Nega-tive	False positive	**False positive rate**	True positive	**Detec-tion rate**
217	26	191	34	**18%**	26	**100%**

Table 5: Results of the experimentation.

[3] Jaynes C, Webb S, Steele R and Xiong Q, "An open development environment for evaluation of video surveillance systems", 3rd Int. Workshop on PETS.

[4] B Georis, X Desurmont, D Demaret, S Redureau, JF Delaigle, B Macq, "IP distributed computer-aided video-surveillance system", Université Catholique de Louvain, Multitel ASBL, Belgium , Intelligent Distributed Surveillance Systems, IEE Visual Information Engineering Professional Network, 26 February 2003.

[5] Chris Stauffer, "Adaptive Background mixture models for real-time tracking", W.E.L Grimson, The Artificial Intelligence Laboratory, Massachusetts Institute of Technology , Cambridge,MA 02139.

[6] lngemar J. Cox and Sunita L. Hingorani "An Efficient Implementation of Reid's Multiple Hypothesis Tracking Algorithm and Its Evaluation for the Purpose of Visual Tracking".

[7] Justus H.Piater, Stéphane Richetto, and James L. Crowley "Event-based Activity Analysis in Live Video using a Generic Object Tracker", , Projet Prima, Laboratoire GAVIR-IMAG, INRIA Rhône-Alpes, Proceeding 3rd IEEE Int. Workshop on PETS, Copenhagen, June 1 2002.

[8] Tim 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.

[9] P. Correia; F. Pereira; "A Proposal for the Classification of Video Segmentation Application Scenarios", Proc. Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS), London, United Kingdom, pp. 188 - 195, April 2003

[10] "Alarming false alarms", CCTV TODAY, July/August 2003.

[11] Kyungnam Kim, Thanarat Horprasert, David Harwood, Larry Davis, "Codebook-based Background Subtraction and Performance Evaluation Methodology".

[12] Adam Milstein, Javier Nicolas Sanchez, Evan Tang Williamson, "Robust Global Localization Using Clustered Particle Filtering".