A Video System for Urban Surveillance: Function Integration and Evaluation

R. J. Oliveira
IST
zer@netcabo.pt

P. C. Ribeiro
IST-ISR
pribeiro@omni.isr.ist.utl.pt

J. S. Marques
IST-ISR
jsm@isr.ist.utl.pt

J. M. Lemos
IST-INESC-ID
jlml@inesc.pt

Abstract: This paper is concerned with video sequence analysis for urban area surveillance applications. The aim is to detect, track and classify targets entering a urban scene under varying illumination conditions and distracters. The paper contributions consist in the integration of algorithms for performing the various tasks and in their statistical evaluation. Results are presented on the basis of a benchmark video sequence.

1 Introduction

Video sequence analysis for surveillance applications has been the subject of several recent research papers [6, 8, 14, 2, 4]. The system described in most of these works comprise the functions of object detection, tracking, recognition and classification. High level tasks such as object activity recognition has also been addressed , e.g., [8].

The problem of object detection has been tackled using statistical models of the background image [15, 8, 14, 11], frame differences [1] techniques or a combination of both [13]. Several techniques have also been used for object tracking in video sequences in order to cope with multiple interacting targets. These range from simple solutions such as heuristic methods or Kalman trackers, based on the nearest neighbor rule [17], to multiple hypothesis trackers such as the PDAF [17], multiple hypothesis tree [7] or long term tracking using Bayesian Networks [2]. Additional features such as color can also be used to enhance the tracking performance [4, 10].

Object recognition and classification is performed using statistical Pattern Recognition and neural networks methods. Several features, which explore the specific condition of the problem can be used. These include geometric features such as bounding box aspect ratio, motion patterns and color histogram [14, 8].

Most of the existing literature address the above problems by considering algorithms for their solution without concern for their evaluation. Evaluation problems have been considered in [16, 5]. These tackle mainly segmentation problems of video processing algorithms aiming at object representation and coding. In [5], ROC curves for performance evaluation of video surveillance processing system are considered. However, only object detection is studied. Furthermore only two types of errors (false alarms and miss detections) are treated, nothing being said about other types of errors, such as region splitting and region merging, which degrade the performance of the overall system. Therefore, the main contribution of this paper consists in the statistical evaluation of the methods for each of the surveillance tasks. The system considered results from the integration of modules performing the tasks of object detection, classification and recognition.

This paper is organized as follows. After a brief state of the art review which motivates this paper contributions (this section), the surveillance system considered is described in section 2. Experimental evaluation is then performed in section 3. Conclusions are presented in section 4.

2 System Description

The surveillance system implemented can be viewed as four independent, but interacting modules: detection, tracking, classification and recognition. The figure 1 describes the system and the interaction between its modules. To perform the detection task, a robust real-time algorithm, suggested by T. Boult [15] was adapted. The approach followed uses two adaptive background images, per-pixel adaptive thresholds and a region grouping algorithm, named quasi-connected components(QCC).

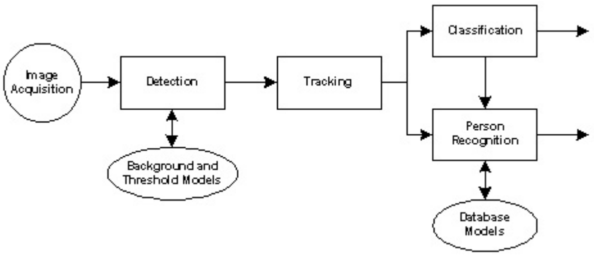


Figure 1: System Block Diagram.

The tracking algorithm determines the overlap between detected regions in consecutive frames, in order to link them, when no ambiguity exists. The linking of an active region in consecutive frames originates a stroke, which describes the evolution of the mass center over time.

The classification task is performed each frame for all active regions detected, and the classification of a stroke is performed by determining the most voted class.

To cope with tracking ambiguities, a color-based recognition module is also integrated in the system.

In order to achieve real-time capability, the detection module, which is the most time-consuming one, was implemented in C using the *Intel Integrated Performance Primitives* [9]. The other modules were implemented in *Matlab* R13, which allow the final system to operate over 30 fps with 768x576 images, using an *Intel Centrino* 1.3 running *Windows XP*.

2.1 Detection

Like many systems, our processing starts with a change-detection method based on background subtraction. The fact that undetected targets cannot be tracked, makes detection a crucial stage. The main difficulties of such approach lie in the fact that, even in controlled environments, the background undergoes a continual change, mostly due to the existence of lighting variations and distractors (example:clouds passing by, branches of trees moving with the wind). Target occlusion and interaction with the scene rises additional problems. To overcome these difficulties, the robust and fast algorithm described in [15] was implemented. The robustness towards lighting variations of the scene is achieved using adaptive background models and adaptive per-pixel thresholds. The use of multiple backgrounds and the grouping technique QCC contribute to the robustness of the algorithm towards unwanted distractors.

The system implemented uses two gray scale background models B_1 and B_2, created during a training phase. The idea is to have both a lower and a higher pixel value, contemplating this way the variations of "non-target" pixels in the scene. The per-pixel threshold, T_L (low threshold), is then initialized to be above the difference between the two backgrounds. A higher threshold T_H is also created, resulting from the addition of a constant value (which represents the sensitivity of the algorithm) to the threshold T_L.

2.2 Tracking

The purpose of tracking is to determine the spatio-temporal information of each target present in the scene, thereby constructing its stroke. Since the visual motion of targets is always small in comparison to their spatial extends, no position prediction is necessary to construct the strokes [14].

The approach followed associates the active regions of the present frame I^t, with the regions in the previous frame I^{t-1} by region overlap. Five different situations were considered, namely: target entering or leaving the scene, a merge or a split between multiple targets or a match. A unidirectional matching matrices method is used to determine what action is to be taken to a stroke. Whenever there are interactions between multiple targets (splits and merges), the old strokes are interrupted and a new one is created. If there is a match, the stroke is updated. When a target enters the scene a new stroke is started, when one leaves the scene, its stroke is discontinued. This method allows posterior processing, combining the information gathered in the strokes to obtain the full path of a target in the scene.

Tracking also interacts with the detection, since the space over time information is known. When a target stops in the scene for a certain amount of time, the tracker has the possibility to merge this target in the background if it no longer finds it interesting (a vehicle that parks).

2.3 Classification

For the classification task three main questions must be answered, namely: which classes should be considered, which features best separate these classes and which classifiers best adapt to the previous choices?

One of the main goals of the classifier is to achieve low miss-classification probabilities while considering a wide spectrum of classes. At the same time the goal was not to consider time-dependent features, limiting the classifier exclusively to geometric properties. In this way the resulting classifier can be used in different machines, as it is independent of the achieved frame-rate. The Pets2001 Dataset (Camera1 and Camera2) is set as being a typical working situation, thus the classes considered are: one person, two persons, three persons, one vehicle, two vehicles and mixed groups. The mixed groups consider all other combinations of people and vehicles.

A R^3 feature space was chosen. Two of the chosen characteristics are measurements which take into account the size of the target and the size of its bounding box in pixels.

The third measurement, the *normalized size*, represents how big or small a target is, in comparison with a single person in the same zone $\mathcal{Z}$ of the scene.

$$F_1 = \frac{\text{height}}{\text{width}} \qquad F_2 = \frac{\text{target area}}{\text{bounding box area}}$$

$$F_3 = \frac{\text{target area}}{\text{single person area in } \mathcal{Z}}$$

The use of this feature assumes that the relation between the normalized mean size of each class, is constant. Experimental testing revealed that this assumption was acceptable and the overall results were improved about 10%. A training process is necessary to establish the typical size of a person in each zone of the image [12].

Figures 2 and 3 show two different views of the R_3 feature space, where each dot represents a manually classified target into the chosen classes.

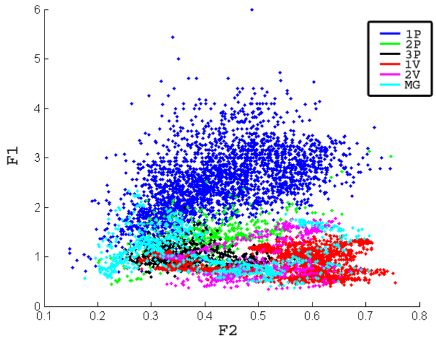


Figure 2: F_1 and F_2 scatter diagram.

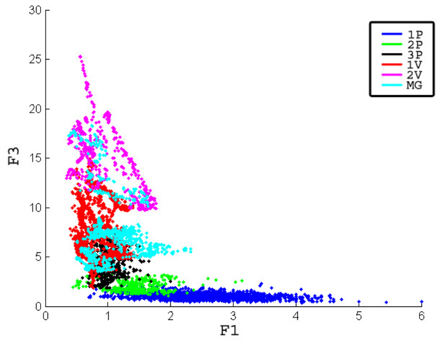


Figure 3: F_1 and F_3 scatter diagram.

The classes that comprise several merged targets, cannot be described by a gaussian distribution over the feature space. These can assume many different configurations, which makes them harder to parameterize. This suggests the choice of a non-parametric classifier, for example the K-Nearest Neighbors algorithm (KNN), which directly estimates the *a posteriori* probability $P(\omega_i|x)$, using adaptive Parzen Windows.

The classification task interacts with the tracker in each frame, voting for the class of each detected target. In this way, a final class is chosen for each stroke as being the most voted one.

2.4 Recognition

As in the classification module, no time information is used to perform the recognition task. This recognition process is aimed at recognizing in a short term period, i.e. targets that become occluded for a few seconds or targets that merge for a few seconds and then split again. The models are characterized by the *pdf* estimates of the chosen feature space, in this case color. The features representing the model are a color histogram in *RGB* color space, comprised of 10 bins of R correlated to 10 bins of G, concatenated with 10 bins of B, giving a total of 110 bins.

The model and the candidate model are represented as follows:

$$\begin{array}{lll} \text{model:} & \hat{\mathbf{q}} = \{\hat{q}_u\}_{u=1\ldots m} & \sum_{u=1}^{m} \hat{q}_u = 1 \\ \text{candidate:} & \hat{\mathbf{p}} = \{\hat{p}_u\}_{u=1\ldots m} & \sum_{u=1}^{m} \hat{p}_u = 1 \end{array}$$

For the error measure between histograms the Euclidean distance was used.

3 Experimental Evaluation

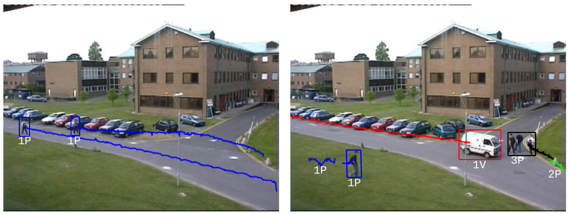
To evaluate the performance of the different tasks, the correct results of each task are manually constructed. For the detection task a ground truth was created (Pets2001 Camera1 training and testing sequences) and an automatic method, similar to the matching matrices, decides if a detection falls into one of the following cases: correct detections, false alarms, miss detections, splits and merges.

For the classification task, the detected targets were manually classified and are distributed by the different classes as follows: 2652 isolated persons, 411 groups of two people, 607 groups of three people, 2998 isolated vehicles, 547 groups of two vehicles and 829 mixed groups.

For isolated persons, seven persons are tracked and a reference random model chosen for each one, in order to test the recognition methods. The evaluation procedure is done offline comparing the models of the detected persons with the reference models of all the persons previously seen.

3.1 Working Examples

Below, two working examples of the surveillance algorithm are shown, in which the color represents the class of the objects, as in figures 2 and 3. The color of the box around the target represents the voted class in the present frame. The most voted class for each target is represented by the line color and the text below stroke.



(a) (b)

Figure 4: Working examples

Figure 4b shows several targets being followed simultaneously. The group of three persons was correctly classified as being a group of two at first, because until then, the rightmost element was still out of the scene, and afterwards as a group of three. The path of the leftmost person shows more than one stroke, due to a detection error, in this case a splitting. As explained in section 2.2 these situations make the tracker start new strokes. These strokes can be linked together using the recognition algorithm and some heuristics.

The figure 5 represents the end result of the *Pets2001* sequence analysis, showing all targets path and class.

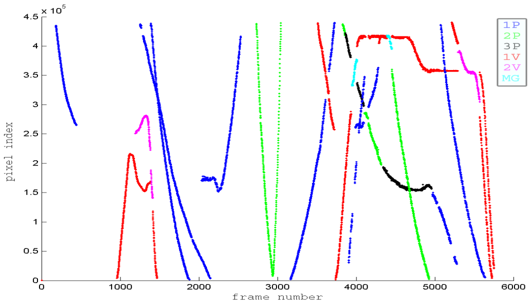


Figure 5: Stroke map obtained in Pets2001 Camera 1, training and testing sequences

As groups are the most difficult to classify, figure 6 shows four examples of detected groups and how they enter the feature extraction procedure (2nd row). Due to partial occlusion the rightmost example was misclassified as being a group of two persons.

3.2 Statistical Evaluation

The statistical evaluation of a detection algorithm can be done by determining its Receiver Operating Characteristics (ROC) curves [5] (figure 7). The merge or split per frame rate can also be calculated, providing further information about the algorithms characteristics (figure 8). In some applications merge or splits can be very undesirable, for example when used for tracking and classification.

The examples shown in figures 7 and 8 show how the parameter sensibility defined in chapter 2.1 influences the operation of the detection algorithm. The study of other parameters, further contribute to the use of the full potential of every such algorithm.

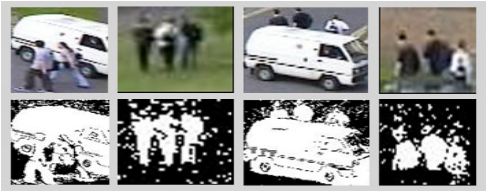


Figure 6: Detection and classification examples: the rightmost example was incorrectly classified as 2P, the other examples were correctly classified as MG, 3P, MG.

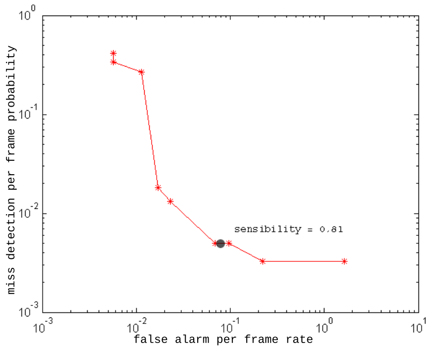


Figure 7: ROC curve for pairs ($p_{detection}$,r_{fa})

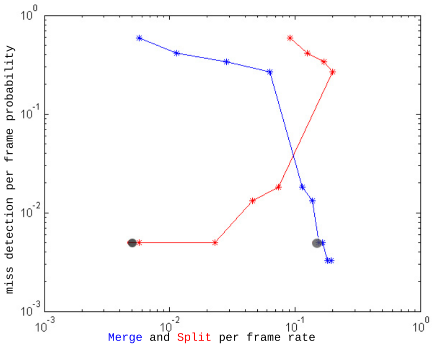


Figure 8: ROC curve for pairs ($p_{detection}$,r_{merge}) and ($p_{detection}$,r_{split}).

Target classification and recognition was evaluated using the confusion matrix and the error probability. Two classifiers were considered: the Bayes classifier assuming that the data is Gaussian and the K-NN classifier. Comparing the confusion matrices of each classifier, two relevant aspects can be compared. First, the inclusion of a given feature can be tested, to determine if it is relevant to differentiate the chosen classes. Second, testing different algorithms for the same feature space helps to determine which one is best suited. The table 1 can be seen as the compilation of both types of tests. The confusion matrices (tables 2 and 3) show a detailed result of the classifier and help to determine which are the most difficult classes to classify for every configuration of algorithm and features. For example, for the Bayes classifier, the class *Mixed Groups* is where it fails the most. These groups include many different configurations of people and vehicles and also many different poses for its elements, leading clearly to a non gaussian distribution.

Table 1: Comparison of error probability with different classifiers and features

	Bayes	KNN
Without F_3	37 %	18%
With F_3	19%	6%

Table 2: Confusion matrices for the Bayes classifier using features: F_1, F_2 and F_3

	1P	2P	3P	1V	2V	MG
1P	**95.8**	4.4	0	0.0	0	0.4
2P	4.1	**84.7**	10.5	0	0	0.7
3P	0	6.6	**86.3**	3.5	0	3.6
1V	0	0.3	23.3	**71.5**	3.3	1.7
2V	0	0	0	0.6	**97.8**	1.7
MG	0	0	17	15.7	18.1	**49.2**

Table 3: Confusion matrices for the KNN classifier using features: F_1, F_2 and F_3

	1P	2P	3P	1V	2V	MG
1P	**99.3**	0.8	0	0	0	0
2P	8.0	**87.4**	4.1	0.5	0	0
3P	0	3.8	**78.3**	10.9	0	7.1
1V	0.0	0.2	1.1	**96.6**	0.8	1.2
2V	0	0	0	4.6	**90.1**	5.3
MG	0	0	5.9	8.9	4.1	**81.1**

Although the recognition task is only used in specific occasions: e.g., in the case of entering targets, temporary occlusion or target merge and split, the testing was done in every frame. The table 4 shows a confusion matrix for the seven persons considered in the Pets2001 Camera1 Data set, where an error probability of 18% is achieved . As with the classification algorithm, these tables help determine which features, or in this particular case, which color histogram, best suits this problem.

Table 4: Confusion matrices for RG+B color histograms

	P1	P2	P3	P4	P5	P6	P7
P1	**89,0**	0	0	0	0	11,0	0
P2	13,1	**82,5**	4,5	0	0	0	0
P3	0	10,3	**89,8**	0	0	0	0
P4	5,4	52,3	0	**42,3**	0	0	0
P5	0	2,7	0	0	**97,3**	0	0
P6	0	0	0	58,8	0	**41,2**	0
P7	0	0	0	0	9,4	3.6	**87,0**

4 Conclusions

Evaluation of an integrated real-time video system for urban surveillance has been considered. This system comprises the modules of object detection, classification and recognition. The main contribution of the paper consists in the statistical evaluation of tasks performed by these modules. A benchmark video sequence has been used for the application. The evaluation is based on the comparison of the system output with the ground truth obtained by manually editing the video sequence. Several types of errors were taken into account. The procedure developed can be used as a systematic methodology for video surveillance evaluation.

References

[1] A. Lipton, et al. ”Moving Target Classification And Tracking from Real-time Video”,*IEEE WACV*, 1998, pp.8-14.

[2] A. Abrantes, J. Marques, J. Lemos, ”Long Term Tracking Using Bayesian Networks”, *in IEEE ICIP*, 609-612, 2002.

[3] A. Cavallaro, et al. ”Objective Evaluation Of Segmented Quality Using Spatio-Temporal Context”.*in Proc. of IEEE ICIP*, 22-25 September 2002.

[4] D. Comaniciu, et al. ”Kernel-Based Object Tracking”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 4, April 2003.

[5] F. Oberti, et al. ”ROC curves for performance evaluation of video sequences processing systems for surveillance applications”, DIBE-University of Genoa.

[6] G. Foresti, C. Regazzoni. ”New Trends in Video Comunications, Processing and understanding in Surveillance Applications”, *in ICIP 2001*, October 7, 2001, Thessaloniki.

[7] I. Cox, S. Hingorani, ” An Efficient Implementation of Reid’s Multiple Hypothesis Tracking Algorithm and its Evaluation for the Purpose of Visual Tracking”, IEEE PAMI, 138-150, 1996.

[8] I. Hariataoglu, et al. ”W4: A Real-Time Surveillance of People and Their Activities”, *IEEE PAMI*, 22, No. 8, August 2000.

[9] Intel Corporation, *Integrated Performance Primitives for Intel Architecture*, Reference Manual, U.S.A., 2002. (http://developer.intel.com)

[10] P. Prez, et al, ”Color-Based Probabilistic Tracking”, *European Conference on Computer Vision 2002*, LNCS 2350, pp. 661-675, 2002.

[11] R. Cucchiara, C. Grana, A. Prati, ”Detecting Moving Objects and their Shadows: an evaluation with the PETS2002 dataset” *IEEE ECCV, 18-25, 2002.*

[12] R. Oliveira, P. Ribeiro, J. Marques, J. Lemos. ”A Video System for Urban Surveillance: Function Integration and Evaluation”.*IST Internal Report*, 2003.

[13] R. Collins, et al. ”A System for Video Surveillance and Monitoring”, CMU-RI-TR-00-12, 2000.

[14] S. McKenna, et al, ”Tracking Groups of People”, *CVIU* 80, 42-56 (2000).

[15] T. Boult, et al. ”Into the Woods: Visual Surveillance of Noncooperative and Camouflaged Targets in Complex outdoor Settings”, *in Proceedings of the IEEE*, vol 89, NO 10 October 2001.

[16] Y. Zhang, ”A Survey on Evaluation Methods for Image Segmentation”, *Pattern Recognition, Vol 29. No. 8,pp.1335-1346*, 1996.

[17] Y Bar-Shalom, T. Fortmann, ”Tracking and Data Association”, Academic Press, 1988.

Acknowledgement: This work was partially suported by *FCT* under project *Long Term Tracking*. The authors would like to thank Dr. Jacinto Nascimento for helpfull suggestions concerning error types.