A Real-Time System for Classification of Moving Objects

E. Rivlin, M. Rudzsky, R. Goldenberg, U. Bogomolov, S. Lepchev
Computer Science Department
Technion - Israel Institute of Technology
32 000 Haifa, Israel

Abstract

The paper describes a system for moving object classification. Being restricted by real-time system constraints we found a small set of features, characterizing object shape and motion dynamics. The system was tested on a large movies database including more than 100 images sequences showing people, animals, vehicles and plants in motion. The SVM classifier [1] was used in our system, yielding very good classification results.

1 Introduction

We developed a software system which is capable, given a sequence of frames acquired by a static camera, to detect and track moving objects. The system extracts static and dynamic characteristic features of moving objects and uses them to distinguish between several predefined object categories. In what follows we describe the main components of the system including: initialization and adaptive update of a background model, detection and tracking of moving objects, extraction of feature vectors, and the classification.

2 Change Detection and Background Modelling

Background subtraction and temporal differencing of consecutive frames are popular methods for change detection in many applications such as object tracking [15], intruder detection [4], traffic monitoring [8], inter-frame data compression [12] and others. Temporal differencing is adaptive to changes in the environment, but does not detect the entire object. On the other hand, the background subtraction can provide more reliable information about moving objects, but it requires more complex processing for adaptation of the background to changes in lighting conditions. It may also lead to "holes" when stationary objects attributed to the background start to move. Therefore, in some works [2] a hybrid approach is applied.

While global thresholding is the simplest method for change detection, it can be improved by local thresholding, particularly when the scene illumination varies locally over time. Noisy difference maps can be much improved by removing small isolated change pixels, merging close regions of change, incorporating connectivity, and performing hysteresis thresholding [13].

Another way to improve the change detection scheme is to build a better background model. The background adaptation methods vary from monochromatic filtering [8], [6] and using various color spaces [7] to statistical background models [14]. A review on background subtraction in video surveillance systems can be found in [10].

2.1 Background Initialization

Creation of an initial background model is an important yet not solved problem. There are only a few works dealing with background initialization. The general assumption that the background can be extracted by using the scene without moving objects is not always valid for outdoor sequences. A method for background initialization for a sequence containing foreground objects is presented in [5]. It uses the following assumptions, which we also adopt here: each pixel in the image will reveal the background for at least a short interval in the sequence; the background is approximately stationary, only small background motion may occur; and a short processing delay is allowed subsequent to acquiring the training sequence. In our algorithm the background initialization is done in the first 1 - 2 seconds of the processing, when the background model is learned by the system.

First we initialize the background image $B(i,j)$ by the first frame $B(i,j) = I_1(i,j)$ and create a binary mask M_1 by thresholding the difference between the two consecutive frames I_1 and I_2. In color images we use the Eucledian metric for measuring the distance between the pixel colors.

The binary mask M_1 shows suspicious regions that may result from several factors: changes in illumination, chaotic motion in background (trees or bushes), or be caused by

1051-4651/02 $17.00 © 2002 IEEE

moving objects. In the latter case, these are the foreground pixels. Thus, the suspicious pixels should be traced till their color becomes stable and then we can add them with their color values to the background image.

This is done by looking at the binary mask M_n created by thresholding the difference between every two consecutive frames I_{n-1} and I_n and updating the background pixels as follows:
if $M_1(i,j) = 1$ *and* $M_n(i,j) = 0$
then $B(i,j) = I_n(i,j); M_1(i,j) = 0$.
The binary masks M_i are filtered using the morphological opening. The process stops when the number of remaining suspicious pixels $N_s = |\{M_1(i,j) = 1\}|$ either decreases below the predefined threshold, or becomes stable. The reason for the latter case can be a sporadic motion in the background (e.g. plants, clouds, etc.). Then, after the process is finished the uninitialized background pixels get their values from the last frame: *if* $M_1(i,j) = 1$ *then* $B(i,j) = I_n(i,j)$. It is clear that this is not a "true" solution, but it is the best we can achieve at this stage. Figure 1 presents different stages of the background image initialization. One can see how the suspicious region initially occluded by a moving car gradually disappears as the car moves away.

2.2 Background Adaptation

An adaptive update of the background is a desirable feature due to the two main reasons: the changes caused by moving objects and the stochastic motion of plants, or changes due to illumination fluctuations. Here we use a simple, yet efficient background update model [9],[2],[13]:
if $M_n(i,j) = 0$
then $B_{n+1}(i,j) = c * B_n(i,j) + (1-c) * I_n(i,j)$
else $B_{n+1}(i,j) = B_n(i,j)$,
where $B_n(i,j)$ is the current background image, $I_n(i,j)$ is the current frame, and $0 < c < 1$ is a predefined constant.

3 Target Detection and Tracking

Target detection is performed using the background subtraction. The difference image is processed by morphological filters and then every sufficiently large connected component is assumed to be a moving object and enters the active objects database. The database stores a number of attributes associated with every moving object. The attributes are used for tracking the object in the subsequent frames and include area, color table, speed and direction of motion.

The objects resulting from a sporadic motion in the background, such as a tree branch motion, are rejected by thresholding the minimal distance travelled by object center of mass during a predefined period of time.

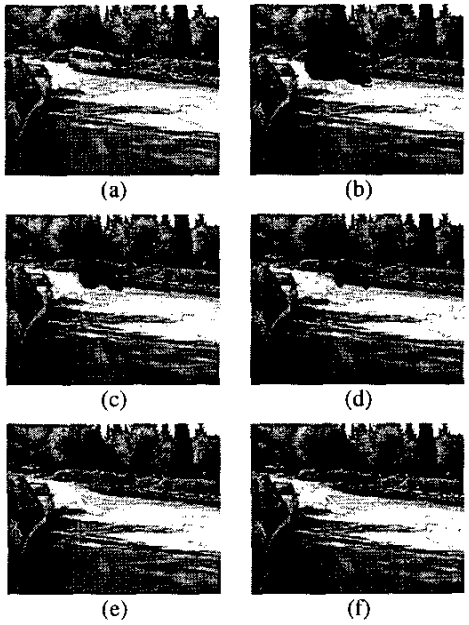


Figure 1. Creation of the background model: (a-e) The current backgound model for the frames 1, 7, 12 and 22 respectively; (f) The final background image.

The color table is a simplified representation of object color data. The target is partitioned into horizontal strips and the color table entries hold the average RGB values for each strip. The number of strips depends on target size. Figure 2 shows an example of the color map for a walking man object. The color table is used for defining an individual threshold for each pixel and every target. It is set for each one of the three color channel to:

$$T(i,j) = \min\{abs(B(i,j) - CT(k)) | k = 1, ..., k_c\},$$

where $T(i,j)$ is the thresholds image for every object, $B(i,j)$ is the background image, $CT(k), k = 1, ..., k_c$ is the target color table of size k_c.

The tracking process is similar to the detection stage, while the processing is done separately for every active target in the database. The search for a new target position is performed in the area predicted by the optical flow computed for the object contour points in the previous frame. The background subtraction uses the thresholds image associated with the object and the new target position is found by choosing the connected component that best matches the object attributes stored in the database.

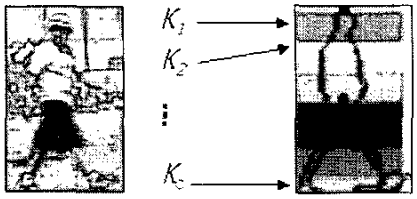


Figure 2. (a) Moving object bounding box (b) The color map

An additional processing includes finding and uniting detached object parts using the stored target template and removing shadows by geometric and color based filters [3],[7],[11].

The object entry in the database is updated with the newly found target data in the current frame. The object is removed from the database if it is not detected for a number of frames or when it leaves the field of view.

The mutual occlusion by the moving objects is predicted by analyzing the targets geometry and relative velocities and the processing is suspended till the occlusion is over. The objects are then found in their predicted positions based on the history data.

Figure 3 presents several examples of detection and tracking. Original frames showing moving vehicle, dog and humans are shown together with the detected contours. The last example shows the tracking of multiple objects.

While tracking, the system accumulates the data required for classification based on the current appearance of the target and its dynamic behavior.

4 Feature Vectors and Classification

Our system performs classification of moving objects into several predefined classes: 'human', 'animal' and 'vehicle'. The 'human' class is further subdivided into 'walking human' and 'running human'.

In supervised classification problem the goal is to assign an unseen pattern to its respective class based on previous examples from each class. In our experiments we decided to use the SVM classifier due to its high speed of learning and reliability in classification.

The feature vector for classification purposes is created from the features, which are most descriptive and represent both static and dynamic properties of moving objects extracted during object tracking. Initially we started from a wide set of features describing the appearance of object contour and its temporal behavior. By experimenting with various subsets of these features we obtained a sufficient reduced feature set, leading to the best classification accuracy.

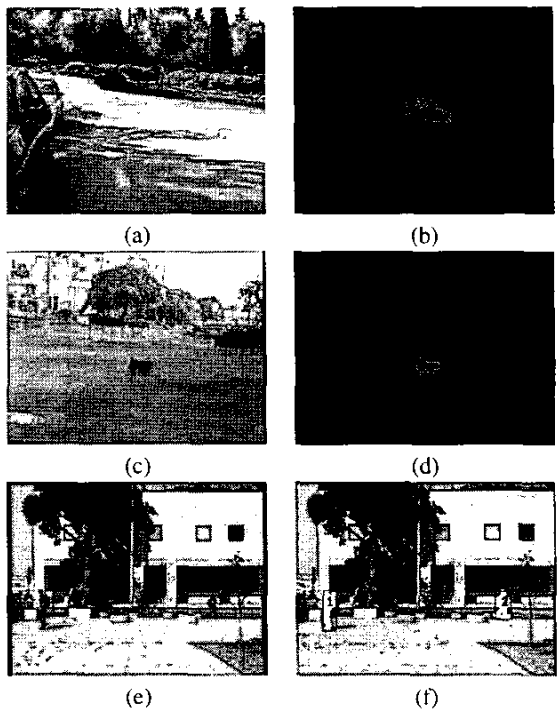


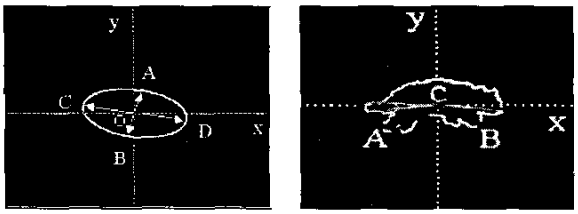
Figure 3. Target detection and tracking: Left column - original frames; Right column - extracted targets.

For example, it turned out that the shape features can improve classification when used in conjunction with other features but have limited power by themselves. The compactness (the ratio between the perimeter squared and the area), which is commonly used as the object shape characteristic, proved to be very non robust due to the perimeter sensitivity to noise and was discarded. Instead, the ratio between the lengths of the 'vertical' and 'horizontal' axes of the ellipse fitted to the object contour was used.

The other features providing the best inter-class separation are based on geometric properties of the fitted ellipse (see Figure 4(a)) and the 'star skeleton' (see Figure 4(b)).

The principal angle of the fitted ellipse, i.e. the $\angle DOX$ (Figure 4(a)) - the angle between the OX axis and the "horizontal" (closest to the OX) symmetry axis CD of the ellipse, is taken as one of the components of the feature vector.

Star skeleton is created by connecting the center of mass of the moving object with contour points corresponding to



(a) (b)

Figure 4. Feature extraction: (a) Fitted ellipse (b) Star skeleton

the local maxima of the function measuring the distance between the contour and the center of mass. The feature extracted from the star skeleton is the angle between its 'legs' ($\angle ACB$ in the Figure 4(b)).

To describe the dynamics of motion we include the temporal characteristics of the features above. Namely, the amplitudes of the first two strongest (besides DC) peaks of the Fourier transform of the feature time series extracted from 30 consecutive frames. In addition we take the amplitude of feature value variations during the measurement period.

The SVM classifier we used in our system is the publicly available SVM program from [1].

The classification is performed for every 30 consecutive frames. For each object the classification results are accumulated in the corresponding bin of its classification histogram, which has three bins according to the predefined classes: "human", "animal" and "vehicle". The decision about the object identity is made by voting according to the results accumulated in object's histogram.

We tested the system on a set of 104 movies. The set was randomly split into the training set (10%) and the testing set (the rest). The table below presents the average confusion matrix obtained from 100 such experiments using the feature set described above:

	VEHICLE	ANIMAL	HUMAN
VEHICLE	0.987	0.013	0
ANIMAL	0.049	0.949	0.002
HUMAN	0	0.038	0.962

5 Future work

The system now allows to process 20 frames/sec on Pentium III, 800 MHz. We hope that a code optimization may further improve the performance. We plan to extend the number of supported classes by adding additional features and to make the system more robust to occlusions of any type.

References

[1] C. C. Chang and C. J. Lin. Training nu-support vector regression: theory and algorithms. *Neural Computation*, 2, 2001.

[2] R. T. Collins, A. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. E. O, and Hasegawa. A system for video surveillance and monitoring: Vsam final report. technical report CMU-RI-TR-00-12, Robotics Institute, Carnegie Mellon University, May 2000.

[3] A. Elgammal, D. Harwood, and L. Davis. Non-parametric model for background subtraction. In *ICCV'99*, 1999.

[4] T. J. Ellis, P. Rosin, and P. Golton. Model-based vision for automatic alarm interpretation. *IEEE Aerospace and Electronic Systems Magazine*, 6(3):14–20, 1991.

[5] D. Gutchess, M. Trajkovic, E. Kohen-Solal, D. Lyons, and A. K. Jain. A background model initialization algorithm for video surveillance. In *Proceedings of the Eighth International Conference on Computer Vision*, pages 733–740, Vancouvier, Canada, July 9-12 2001.

[6] I. Haritaoglu, D. Harwood, and L. Davis. W4: Who, when, where, what: A real-time system for detecting and tracking people. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 222–227, 1998.

[7] T. Horprasert, D. Harwood, and L. Davis. A statistical approach for real-time robust background subtraction and shadow detection. In *ICCV'99 Frame Rate Workshop*, pages 1–19, 1999.

[8] D. Koller, J. Weber, and J. Malik. Robust multiple car tracking with occlusion reasoning. In *Proceedings of ECCV*, pages 189–196, 1994.

[9] A. J. Lipton, H. Fujiyoshi, and R. S. Patil. Moving target classification and tracking from real-time video. In *IEEE Workshop on Applications of Computer Vision (WACV)*, pages 8–14, Princeton NJ, October 1998.

[10] A. McIvor, V. Zang, and R. Klette. The background subtraction problem for video surveillance systems. In *International Workshop Robot Vision 2001, Auckland, New Zealand, February 2001. Springer Lecture Notes in Computer Science 1998*, pages 176–183, 2001.

[11] S. J. McKenna, S. Jabri, Z. Duric, A. Rosenfeld, and H. Wechsler. Tracking groups of people. *Computer Vision and Image Understanding*, 80:42–56, 2000.

[12] K. N. Ngan. *Advanced Video Coding: Principles and Techniques*. Elsevier, 1999.

[13] P. L. Rosin. Thresholding for change detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 274–279, 1998.

[14] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 246–252, Fort Collins, Colorado, 1999.

[15] Y.H.Yang and M. D. Levine. The background primal sketch: An approach for tracking moving objects. *Machine Vision and Applications*, 5:17–34, 1992.