Object Classification and Tracking in Video Surveillance

Qi Zang and Reinhard Klette

CITR, Computer Science Department, The University of Auckland
Tamaki Campus, Auckland, New Zealand

Abstract

The design of a video surveillance system is directed on automatic identification of events of interest, especially on tracking and classification of moving vehicles or pedestrians. In case of any abnormal activities, an alert should be issued. Normally a video surveillance system combines three phases of data processing: moving object extraction, moving object recognition and tracking, and decisions about actions. The extraction of moving objects, followed by object tracking and recognition, can often be defined in very general terms. The final component is largely depended upon the application context, such as pedestrian counting or traffic monitoring. In this paper, we review previous research on moving object tracking techniques, analyze some experimental results, and finally provide our conclusions for improved performances of traffic surveillance systems. One stationary camera has been used.

1 Introduction

Recent research in video surveillance systems is focused on background modelling, moving object classification and tracking. A near-correct extraction of all pixels defining a moving object or the background is crucial for moving object tracking and classification. Major occurrences of moving objects in our data are pedestrians and vehicles. The camera(s) position will affect the selection of an appropriate technique for object tracking. Considering the angle between viewing direction and a horizontal ground plane, this angle is often about 0^o which is horizontal, or 90^o which is vertical. In situations of about horizontal or vertical viewing, researchers typically prefer the use of region based tracking, or of contour or snake tracking techniques, because the shape of the extracted moving object is not expected to change much. This assumption simplifies feature calculations for tracking, and the main problem is that moving object may be occluded by each other, or by stationary objects such as buildings. But in non-vertical and non-horizontal situations which are typical for traffic monitoring systems, the angle between the viewing direction and the ground plane can take any value. If vehicles move fast, then the shape of the vehicle will change rapidly. In this case feature based tracking is required which extends simple shape matching approaches.

The primary goal of this paper is to critically discuss the use of tracking methods in different situations. A second goal is to present a hybrid method in using feature based object tracking in traffic surveillance, and report about its performance. The paper is structured as follows: in Section 2, we discuss existing approaches for tracking moving

objects using different techniques in different situations. Section 3 presents our ideas for moving object tracking. Section 4 discusses our performance experiments, and Section 5 finally informs about the obtained analysis results and gives conclusion.

2 Review of Previous Work

Many applications have been developed for monitoring public areas such as offices, shopping malls or traffic highways. In order to control normal activities in these areas, tracking of pedestrians and vehicles play the key role in video surveillance systems. We classify these tracking techniques into four categories:

Tracking based on a moving object region. This method identifies and tracks a *blob token* or a *bounding box*, which are calculated for connected components of moving objects in 2D space. The method relies on properties of these blobs such as size, color, shape, velocity, or centroid. A benefit of this method is that it time efficient, and it works well for small numbers of moving objects. Its shortcoming is that problems of occlusion cannot be solved properly in "dense" situations. Grouped regions will form a combined blob and cause tracking errors. For example, [11] presents a method for blob tracking. Kalman filters are used to estimate pedestrian parameters. Region splitting and merging are allowed. Partial overlapping and occlusion is corrected by defining a pedestrian model.

Tracking based on an active contour of a moving object. The contour of a moving object is represented by a *snake*, which is updated dynamically. It relies on the boundary curves of the moving object. For example, it is efficient to track pedestrians by selecting the contour of a human's head. This method can improve the time complexity of a system, but its drawback is that it cannot solve the problem of partial occlusion, and if two moving objects are partially overlapping or occluded during the initialization period, this will cause tracking errors. For example, [5] proposes a stochastic algorithm for tracking of objects. This method uses factored sampling, which was previously applied to interpretations of static images, in which the distribution of possible interpretations is represented by a randomly generated set of representatives. It combines factored sampling with learning of dynamical models to propagate an entire probability distribution for object position and shape over time. This improves the mentioned drawback of contour tracking in case of partial occlusions, but increases the computational complexity.

Tracking based on a moving object model. Normally model based tracking refers to a 3D model of a moving object. This method defines a parametric 3D geometry of a moving object. It can solve partially the occlusion problem, but it is (very) time consuming, if it relies on detailed geometric object models. It can only ensure high accuracy for a small number of moving objects. For example, [6] solved the partial occlusion problem by considering 3D models. The definition of parameterized vehicle models make it possible to exploit the a-priori knowledge about the shape of typical objects in traffic scenes. [2].

Tracking based on selected features of moving objects. Feature based tracking is to select common features of moving objects and tracking these features continuously. For example, corners can be selected as features for vehicle tracking. Even if partial occlusion occurs, a fraction of these features is still visible, so it may overcome the partial occlusion problem. The difficult part is how to identify those features which belong to the same object during a tracking procedure (feature clustering). Several papers have been published on this aspect. For example, [10] extracts corners as selected

features using the Harris corner detector. These corners then initialize new tracks in each of the corner trackers. Each tracker tracks any current corner to the next image and passes its position to each of the classifiers at the next level. The classifiers use each corner position and several other attributes to determine if the tracker has tracked correctly.

Besides these four main categories, there are also some other approaches on object tracking. [7] presents a tracking method based on wavelet analysis. A wavelet-based neural network (NN) is used for recognizing a vehicle in extracted moving regions. The wavelet transform is adopted to decompose an image and a particular frequency band is selected for input into the NN for vehicle recognition. Vehicles are tracked by using position coordinates and wavelet feature differences for identifying correspondences between vehicle regions [7]. Paper [3] employs a second order motion model for each object to estimate its location in subsequent frames, and a "cardboard model" is used for a person's head and hands. Kalman models and Kalman filters are very important tools and often used for tracking moving objects. Kalman filters are typically used to make predictions for the following frame and to locate the position or to identify related parameters of the moving object. For example, [13] implemented an online method for initializing and maintaining sets of Kalman filters. At each frame, they have an available pool of Kalman models and a new available pool of connected components that they could explain. Paper [12] uses an extended Kalman filter for trajectory prediction. It provides an estimate of each object's position and velocity. But, as pointed out in [5], Kalman filters are only of limited use, because they are based on unimodal Gaussian densities and hence cannot support simultaneous alternative motion hypotheses. So several methods have also been developed to avoid using Kalman filtering. [5] presents a new stochastic algorithm for robust tracking which is superior to previous Kalman filter based approaches. Bregler [1] presents a probabilistic decomposition of human dynamics to learn and recognize human beings in video sequences. [9] presents a much simpler method based on a combination of temporal differencing and image template matching which achieves highly satisfactory tracking performance in the presence of partial occlusions and enables good classification. This avoids probabilistic calculations.

3 A New Approach

Our approach specifies two subprocesses, the extraction of a (new) moving object from the background and tracking of a moving object.

3.1 Object Extraction from background

Evidently, before we start with tracking of moving objects, we need to extract moving objects from the background. We use background subtraction to segment the moving objects. Each background pixel is modelled using a mixture of Gaussian distributions. The Gaussians are evaluated using a simple heuristic to hypothesize which are most likely to be part of the "background process". Each pixel is modeled by a mixture of K Gaussians as stated in formula (1):

$$P(\mathbf{X}_t) = \sum_{i=1}^{K} \omega_{i,t} \eta(\mathbf{X}_t; \mu_{i,t}, \Sigma_{i,t}) \ . \tag{1}$$

where $\mathbf{X}_t$ is the variable, which represents the pixel, and t represents time. Here K is the number of distributions: normally we choose K between 3 to 5. $\omega_{i,t}$ is an estimate of the weight of the ith Gaussian in the mixture at time t, $\mu_{i,t}$ is the mean value of the ith Gaussian in the mixture at time t. $\Sigma_{i,t}$ is the covariance matrix of the ith Gaussian in the mixture at time t. Every new pixel value $\mathbf{X}_t$ is checked against the existing K Gaussian distributions until a match is found. Based on the matching results, the background is updated as follows: $\mathbf{X}_t$ matches component i, that is $\mathbf{X}_t$ decreases by 2.5 standard deviations of the distribution, then the parameters of the ith component are updated as follows:

$$\begin{aligned} \omega_{i,t} &= (1-\alpha)\omega_{i,t-1} + \alpha & (2) \\ \mu_{i,t} &= (1-\rho)\mu_{i,t-1} + \rho\mathbf{I}_t & (3) \\ \sigma^2_{i,t} &= (1-\rho)\sigma^2_{i,t-1} + \rho(\mathbf{I}_t - \mu_{i,t})^{\mathsf{T}}(\mathbf{I}_t - \mu_{i,t}) & (4) \end{aligned}$$

where $\rho = \alpha \mathrm{Pr}(\mathbf{I}_t|\mu_{i,t-1}, \Sigma_{i,t-1})$. α is the predefined learning parameter, μ_t is the mean value of the pixel at time t, and $\mathbf{I}_t$ is the recent pixel at time t. The parameters for unmatched distributions remain unchanged, i.e., to be precise:

$$\mu_{i,t} = \mu_{i,t-1} \quad \text{and } \sigma^2_{i,t} = \sigma^2_{i,t-1} \; . \tag{5}$$

But $\omega_{i,t}$ will be adjusted using formula: $\omega_{i,t} = (1-\alpha)\omega_{i,t-1}$.

If $\mathbf{X}_t$ matches none of the K distributions, then the least probable distribution is replaced by a distribution where the current value acts as its mean value. The variance is chosen to be high and the a-priori weight is low [13]. The background estimation problem is solved by specifying the Gaussian distributions, which have the most supporting evidence and the least variance. Because the moving object has larger variance than a background pixel, so in order to represents background processes, first the Gaussians are ordered by the value of $\omega_{i,t}/\|\Sigma_{i,t}\|$ in decreasing order. The background distribution stays on top with the lowest variance by applying a threshold T, where

$$B = \mathrm{argmin}_b \left(\frac{\sum_{i=1}^{b} \omega_{i,t}}{\sum_{i=1}^{K} \omega_{i,t}} > T \right) \; . \tag{6}$$

All pixels $\mathbf{X}_t$ which do not match any of these components will be marked as foreground. The next step is to remove shadows. Here we use a method similar to [8]. The detection of brightness and chromaticity changes in the HSV space are more accurate than in RGB space, especially in outdoor scenes, and the HSV color space corresponds closely to human perception of color [4]. At this stage, only foreground pixels need to be converted to hue, saturation and intensity triples. Shadow regions can be detected/eliminated as followings: let $\mathbf{E}$ represent the current pixel at time t, and $\mathbf{B}$ represents the background pixel at time t. For each foreground pixel, if it satisfies the constraints

$|\mathbf{E}_h - \hat{\mathbf{B}}_h| < \mathbf{T}_h$, $|\mathbf{E}_s - \hat{\mathbf{B}}_s| < \mathbf{T}_s$ and $\mathbf{T}_{v1} < \mathbf{E}_v/\hat{\mathbf{B}}_v < \mathbf{T}_{v2}$

then this pixel will be removed from the foreground mask. Parameters of shadow pixels will not be updated. Finally, we obtain the moving objects mask, which is applicable for object tracking.

3.2 Object Tracking and Classification

After obtaining an initial mask for a moving object, we have to preprocess the mask. Normally the mask is affected by "salt-and-pepper" noises. We apply morphological

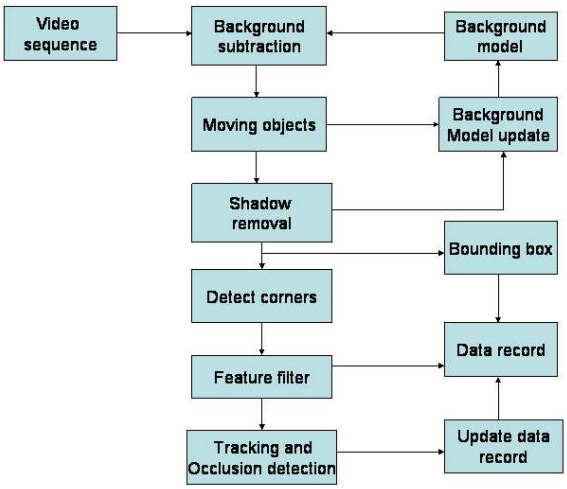

Figure 1: Flow chart sketch of the proposed approach.

filters based on combinations of dilation and erosion to reduce the influence of noise, followed by a connected component analysis for labeling each moving object region. Very small regions are discarded. At this stage we calculate the following features for each moving object region: *bounding rectangle:* the smallest isothetic rectangle that contains the object region. We keep record of the coordinate of the upper left position and the lower right position, what also provides size information (width and height of each rectangle). *color:* the mean R G B values of the moving object. *center:* we use the center of the bounding box as a simple approximation of the centroid of a moving object region. *velocity:* defined as movement of number of pixels/second in both horizontal and vertical direction. In order to track moving objects accurately, especially when objects are partially occluded, and the position of the camera is not restricted to any predefined viewing angle, these features are actually insufficient. We have to add further features that are robust and which can also be extracted even if partial occlusion occurs. From our experiments with traffic video sequences, corners were selected as additional features for tracking. We use the popular SUSAN corner detector to extract corners of vehicles. For each frame, after obtaining a bounding box of the moving object, we then detect corners within the bounding box by applying Susan Quick masks on each pixel. Although sometimes it produces false positives on strong edges, it is faster and can report more stable corners. The corner's position and intensity value is added to a *corner list* of this object. Altogether, the features of a moving object are represented in a five-components vector [bounding box, color, center position, velocity, corner list]. A symbolic flow chart of the proposed method is shown in Figure 1.

3.2.1 Classification of Moving Object Regions

In our captured traffic scenes, moving objects are typically vehicles or pedestrians. We use the ratio of height/width of each bounding box to separate pedestrians and vehicles. For a vehicle, this value should be less than 1.0, for a pedestrian this value should be

greater than 1.5. But we also have to provide flexibility for special situations such as a running person, a long or taller vehicle. If the ratio is between 1.0-1.5, then we use the information from the corner list of this object to classify it as a vehicle or a pedestrian (a vehicle produces more corners). This is a simple way to classify moving objects into these two categories.

3.2.2 Tracking of Moving objects

For moving object tracking we use a hybrid method based on bounding box and feature tracking. During the initialization period a data record is generated for each object: a label for indexing and the five elements of its vector. New positions are predicted using a Kalman filter. For each new frame, the predicted position is searched to see whether it can find any match with the previous data record. If a matching object region is found, it is marked as 'successfully tracked' and belongs to a normal move; if we cannot find any match, then the object may have changed lanes, or stopped, or exceeded the expected speed. So an unmatched object will be checked against already existing objects in the data record. If matched, then it is also marked as 'successfully tracked'; if still not yet matched, it will be marked as a new object and added to the data record. If an existing object is not being tracking for 5 frames, it will be marked as 'stopped'. According to the video capturing speed, we also define a threshold, which is used for marking 'tracking finished'. Matching is performed within certain thresholds for the different feature vector elements. The three main elements used for matching are: same color, a linear change in size, and a constant angle between the line 'corner point-upper left point' versus the line 'corner point-lower bottom point'. Occlusions are reported if bounding boxes are overlapping. In case of partial occlusions, calculated corners and further feature vector elements are tested for making a decision. Finally the data record will be updated using the results of the matching process.

4 Experimental results

Our approach is implemented on a PC under Linux. Different image sequences have been used: highway with heavy traffic, and a road intersection with vehicles and pedestrians. All sequences are captured in daytime. Figure 2 Left shows moving objects together with bounding boxes and centers marked by white crosses. Figure 2 Right shows examples of detected corners marked by white dots. We set the threshold of the corner detector to a higher value, in order to detect and keep only obvious corners, because "unclear corners" are easily lost, which will affect the tracking accuracy. Corners are only detected within bounding boxes, which not only saves computation time, but also simplifies a common feature tracking problem: how to group features belong to the same objects. After corner detection, we use the identified positions and their intensity values. The average number of corners per vehicle is 26. Our hybrid approach has another advantage, which is to allow the calculation of an important attribute: the angle between the line 'detected corner-upper left position of bounding box' and line 'detected corner-lower right position of bounding box'. This angle is very useful for tracking, because the bounding box shrinks or expands while the object moves, but this angle will still remain unchanged. Of course, this reflects our assumption that the viewing area on a road or highway is basically planar and does not change orientation. The image size is 320 x 240, average processing rate is 4-6 frames/second, on average of 0.2 second per frame. The processing times are given in Table 1.

Step	*Average time*
Object extraction	0.105
Feature extraction	0.025
Object tracking	0.07
Total	0.2

Table 1: Average processing times in seconds.

Figure 2: Left: An enlarged picture showing detected corners of vehicles marked by white dots. Right: Bounding boxes of moving vehicles and their centers marked by white crosses.

5 Conclusions

Moving object tracking is a key task in video monitoring applications. The common problem is occlusion detection. In this case the selection of appropriate features is critical for moving object tracking and classification. We propose a hybrid method of both bounding box and feature tracking to achieve a more accurate but simple object tracking system, which can be used in traffic analysis and control applications. Corners are detected only within the bounding rectangle. In this way we reduced computation time and avoided the common feature grouping problem. Corner attribute is very helpful in feature tracking, in our approach we use the stable angle between the line 'detected corner point-upper left point' versus the line 'detected corner point-lower bottom point'. We use the ratio of height/width plus corner information to classify vehicles and pedestrians. This method proved to be easy and efficient, but it only works well on separated regions. So removing shadows is an important preprocessing task [14] for the subsequent extraction of moving objects masks, because shadows merge otherwise separated regions. Future work will also apply 3D analysis (a binocular stereo camera system and an infrared camera), which allows a more detailed classification of cars. The intention is to identify the type of a vehicle. The height value of the car is, for example, easily to extract from the infrared picture.

References

[1] C. Bregler: Learning and recognizing human dynamics in video sequences. In Proc. *IEEE Int. Conf. CVPR'97*, pages 568-574, 1997.

[2] A. Cavallaro, F. Ziliani, R. Castagno, and T. Ebrahimi: Vehicle extraction based on focus of attention, multi feature segmentation and tracking. In Proc. *European signal processing conference EUSIPCO-2000*, Tampere, Finland, pages 2161-2164, 2000.

[3] I. Haritaoglu, D. Harwood, and L. S. Davis: W4: Who? When? Where? What? A real-time system for detecting and tracking people.In Proc. *3rd Face and Gesture Recognition Conf.*, pages 222-227, 1998.

[4] N. Herodotou, K. N. Plataniotis, and A. N. Venetsanopoulos: A color segmentation scheme for object-based video coding. In Proc. *IEEE Symp. Advances in Digital Filtering and Signal Proc.*, pages 25-29, 1998.

[5] M. Isard, and A. Blake: Contour tracking by stochastic propagation of conditional density. In Proc. *European Conf. Computer Vision, Cambridge, UK*, pages 343-356, 1996.

[6] D. Koller, K. Daniilidis, and H. H. Nagel: Model-based object tracking in monocular image sequences of road traffic scenes. *Int. Journal Computer Vision*, **10**:257–281, 1993.

[7] J. B. Kim, C. W. Lee, K. M. Lee, T. S. Yun, and H. H. Kim: Wavelet-based vehicle tracking for automatic traffic surveillance. In proc. *IEEE int. Conf. TENCON'01*, Aug, Singapore, Vol. 1, pages 313-316, 2001.

[8] P. Kaew Tra Kul Pong, and R. Bowden: An improved adaptive background mixture model for real-time tracking with shadow detection. In Proc. *2nd European Workshop Advanced Video Based Surveillance System*, Sept 2001.

[9] A. J. Lipton, H. Fujiyoshi, and R. S. Patil: Moving target classification and tracking from real-time video. In Proc.*IEEE Workshop Application of Computer Vision*, pages 8-14, 1998.

[10] B. McCane, B. Galvin, and K. Novins: Algorithmic fusion for more robust feature tracking. *Int. Journal Computer Vision*, **49**: 79–89, 2002.

[11] O. Masoud, and N. P. Papanikolopoulos: A novel method for tracking and counting pedestrians in real-time using a single camera. *IEEE Trans. Vehicular Technology*, **50**:1267-1278, 2001.

[12] R. Rosales, and S. Sclaroff: Improved tracking of multiple humans with trajectory prediction and occlusion modeling. In Proc. *Workshop on the Interpretation of Visual Motion at CVPR'98*, Santa Barbara, CA, pages 228-233, 1998.

[13] C. Stauffer, and W. E. L. Grimson: Adaptive background mixture models for real-time tracking. *Computer Vision and Pattern Recognition*, **2**: 246-252, 1999.

[14] Q. Zang, and R. Klette: Evaluation of an adaptive compositeGaussian model in video surveillance. In Proc.*Image and Vision Computing New Zealand 2002*, pages 243-248, 2002.