

Robust Appearance Modeling for Pedestrian and Vehicle Tracking

Wael Abd-Almageed and Larry S. Davis

Institute for Advanced Computer Studies
University of Maryland
College Park, MD 20742
{wamageed, lsd}@umiacs.umd.edu

Abstract. This paper describes a system for tracking people and vehicles for stationary-camera visual surveillance. The appearance of objects being tracked is modeled using mixtures of mixtures of Gaussians. Particle filters are used to track the states of object. Results show the robustness of the system to various lighting and object conditions.

1 Introduction

Detecting and tracking moving people and vehicle is a critical task in any visual surveillance system. Without robust detection and tracking, further video understanding tasks such as activity recognition or abnormal activity detection is not possible.

Robust, realtime tracking algorithms must satisfy a few characteristics. First, an accurate appearance model must be estimated for the objects being tracked as well as the background. Second, the appearance models must be parameter-light or preferably nonparametric in order to increase the level of autonomy of the tracker. Finally, the computing the appearance model must be not computationally expensive to facilitate realtime performance.

In this paper we use our previous work on density estimation using mixtures of mixtures of Gaussians [1] to model the appearance of the objects and the background. Tracking the state of the object is achieved using particle filters.

This paper is organized as follows. Section 2 briefly discusses background subtraction as a classic method for detecting moving objects. For more details on this algorithm, the reader is referred to [1]. In Section 3 we discuss appearance modeling using mixtures of mixtures of Gaussians. The particle filter tracker is introduced in Section 4. Results are presented in Section 5 for tracking people and vehicles under different lighting conditions.

2 Moving Object Detection

Detecting moving objects in stationary camera surveillance is classically performed using background subtraction. To build a background image model, I_{BG}

we use a simple median filtering approach as shown in Equation 1

$$I_{BG}(x, y) = \text{median}_{i=1}^{N_{BG}} I_i(x, y) \quad (1)$$

where N_{BG} is the number of images used to model the background. The probability that a given pixel belongs to the moving foreground F is given by Equation 2

$$p((x, y) \in F) = 1 - \exp\left(-\frac{(I(x, y) - I_{BG}(x, y))^2}{\sigma_F^2}\right) \quad (2)$$

where σ_F is a motion-sensitivity system-parameter. Background subtraction is followed by a series of morphological operations to remove noise and very small moving objects. Connected component analysis is then applied to the resulting image in order to find the independently moving objects. The appearance of each object is modeled using the algorithm described in Section 3 and a tracker is instantiated as will be shown in Section 4.

3 Appearance Modeling using Mixtures of Mixtures

Let $Y = \{\mathbf{x}_i\}_{i=1}^M$ be a set of M vectors to be modeled. If we apply the mean-shift mode finding algorithm, as proposed in [2], and only retain the modes with positive definite Hessian, we will obtain a set of m modes $Y_c = \{\mathbf{x}_{c_j}\}_{j=1}^m$ which represent the local maxima points of the density function, where $m \ll M$. For details on computing the Hessian, the reader is referred to [3].

To infer the structure of the data, we start by partitioning Y into m partitions each of which corresponds to one of the detected modes. For all vectors of Y we compute a Mahalanobis-like distance δ defined by:

$$\begin{aligned} \delta(\mathbf{x}_i|j) &= (\mathbf{x}_i - \mathbf{x}_{c_j})^T P_j (\mathbf{x}_i - \mathbf{x}_{c_j}), \\ i &= 1, 2, \dots, M \quad \text{and} \\ j &= 1, 2, \dots, m \end{aligned} \quad (3)$$

where P_j is the Hessian of mode j . The rationale here, as explained in [3] is to replace the covariance matrix by the Hessian which represents the local curvature around the mode \mathbf{x}_{c_j} . Each vector is then assigned to a specific mode according to Equation 4.

$$\mathcal{C}(i) = \arg_j \min \delta(\mathbf{x}_i|j) \quad \text{and } j = 1, 2, \dots, m \quad (4)$$

The data set can now be partitioned as

$$Y = \bigcup_{j=1}^m Y^j \quad (5)$$

where

$$Y^j = \{\forall \mathbf{x}_i \in Y; \mathcal{C}(i) \equiv j\} \quad (6)$$

Each of the detected modes corresponds to either a single Gaussian or a mixture of more than one Gaussian, based on the complexity of the underlying density function. To determine the complexity of density around a given mode \mathbf{x}_{c_j} , we model the partition data \mathbf{Y}^j using a mixture of Gaussians specific to partition j . In other words,

$$p(\mathbf{x}|\Theta^j) = \sum_{i=1}^k \pi_i \mathcal{N}(\mathbf{x}, \mu_i, \Sigma_i) \quad (7)$$

where Θ^j is the parameter set of a k -component mixture associated with mode \mathbf{x}_{c_j} . The initial values for the mean vectors are all set to \mathbf{x}_{c_j} . The initial values for the covariance matrices are all set to \mathbf{P}_j .

Since the structure of the data around \mathbf{x}_{c_j} is unknown, we repeat the process for a search range of mixture complexities $[k_{min}, k_{max}]$ and compute the Penalty-less Information Criterion ($\mathcal{P}IC$) introduced in [1] for each complexity. The mixture that minimizes the $\mathcal{P}IC$ is chosen to represent the given partition.

Applying $\mathcal{P}IC$ to all partitions results in m mixtures of Gaussians with different complexities. The underlying density of the entire data set \mathbf{Y} is now modeled as a *mixture of mixtures of Gaussians* as follows

$$p(\mathbf{x}|\Theta) = \sum_{j=1}^m \omega_j p(\mathbf{x}|\Theta^j) \quad (8)$$

where $\Theta = \{\Theta^j, \omega_j; j = 1, 2, \dots, m\}$ is the set of all parameters. (Note that we extend the notation Θ here.) Finally, the weights of the mixtures ω_j s are computed according to Equation 9.

$$\omega_j = \frac{\sum_{i=1}^M p(\mathbf{x}_i|\Theta^j)}{\sum_{j=1}^m \sum_{i=1}^M p(\mathbf{x}_i|\Theta^j)} \quad (9)$$

There are two advantages of this algorithm. Firstly, the appearance model obtained is in closed-form representation. This enables the tracker to compute the likelihood values in $O(1)$ time per feature vector, which significantly improves the speed of the tracker as will be shown in Section 5. Secondly, the algorithm is totally non-parametric in the sense that it does not need manual setting of any of its parameters, compared to the popular Expectation Maximization model which needs *a priori* setting of the number of mixture components and the initial means and covariances.

The importance of modeling each partition using a separate mixture can be shown by modeling the color density of the human object in Figure 1.a. The estimated mixture of mixtures is shown in Figure 1.b. The green partition represents the colors of the pants. Since the pants area is a smooth, dark blue cluster, only one Gaussian is enough to model that partition. On the other hand, more than one Gaussian (precisely four) are needed to model the underlying density of the shirt area (blue partition) because of the different shades of gray in that area.

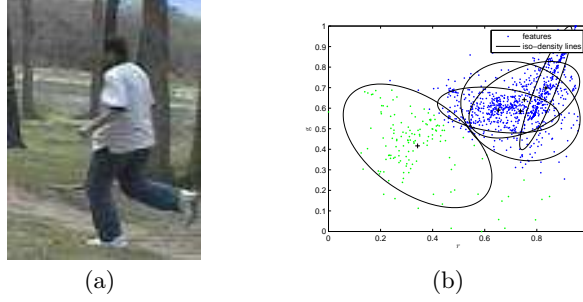


Fig. 1. (a) A moving object and (b) Appearance model of the moving object.

4 Particle Filter Tracking

4.1 Back/Foreground Appearance Models

Background subtraction results in a feature set of background pixels, Y_B , and a number of feature sets representing the detected moving objects, $Y_{O_n}, n = 1, \dots, N$, where N is the number of detected moving objects. These feature sets are used to build an appearance model, $p(x_j|\Theta_B)$, and N appearance models for the detected objects, $p(x_j|\Theta_{O_n})$.

4.2 Particle Filter Tracking

For each of the detected objects, the tracker is formulated as

$$\{s_i^t, \pi_i^t; i = 1, \dots, N_t; t = 1, \dots, T\} \quad (10)$$

where s_i^t and π_i^t represent particle number i at time t and its weight, respectively and $\sum_i^{N_t} \pi_i^t = 1$. N_t represents the number of particles and the subscript t indicates that the number of particles may vary over time and T is the length of the video stream. Each particle represents a combination of translation and scaling of the object being tracked as shown in Equation 11,

$$s_i^t = (\delta_x, \delta_y, \alpha_x, \alpha_y) \quad (11)$$

where δ_x and δ_y represent the translation in the x and y directions, respectively and α_x and α_y represent the scaling in the x and y directions, respectively.

The propagation of the particles follows the state-transition model of Equation 12

$$s_i^t \sim \hat{p}(s_i^{t-1} | s_i^{t-1}, w^{t-1}) \quad (12)$$

where \hat{p} is the probability density function of the states at the current time step and w_{t-1} is the covariance matrix of zero-mean Gaussian process noise. The values of \hat{s}^0 is set to $(0 \ 0 \ 1 \ 1)$ which represents no translation and no scaling and

$\hat{p}(s_i^0)$ is assumed to be uniformly distributed. The four elements of the process noise are assumed to be uncorrelated, normally distributed random variables.

The set of predicted particles $\{s_i^t\}_{i=1}^{N_t}$ corresponds to a set of bounding boxes, $\{B_i^t\}_{i=1}^{N_t}$, on I_t . Each bounding box is evaluated using a Bayesian combination of appearance and motion as shown in Equation 13

$$p(B_i^t | \Theta_O, \Theta_B) = \log \prod_{j=1}^{K_i} \frac{p(x_j | \Theta_O)}{p(x_j | \Theta_B)} \frac{p(x_j \in F)}{1 - p(x_j \in F)} \quad (13)$$

and $i = 1, 2, \dots, N_t$

where K_i is the number of pixels in bounding box i . The bounding box with maximum goodness-of-fit represents the most likely particle which in turn represents the state of the object being tracked at time t as shown in Equation 14

$$\hat{s}_t = \arg_{s_t} \max p(B_i^t | p(x | \Theta_O), \hat{p}(x | \Theta_B)), \quad i = 1, \dots, N_t \quad (14)$$

5 Experimental Results

In this Section, a few number of results is presented on VACE data. The data was processed on a cluster of 15 computers (i.e. nodes) running Linux Operating System. Each node has two 3.0 GHz processors and 8GB of memory.

Figure 5 shows tracking people and vehicles in a night vision surveillance system. From the Figure, we can see that the detector detects moving object that move close to each other as one object and hence tracking is done on the same basis. Also, we can see that detecting new objects and tracking existing ones is performed automatically. Finally, the system does not absorb the white car which comes into the scene and stops indefinitely.

In Figure 5 detects and tracks a single moving object in the video sequence. The performance of the tracker is shown to be very robust against scale changes. The tracker can keep track of the object for long period of time as well as keeping a relatively accurate estimate of the object's scale. Finally, Figure 5 shows another example of detecting and tracking moving people in a daylight camera. The Figure shows that the system can accurate segment the independently moving objects even at very small scales and then track them robustly.

6 Conclusions

In this paper, a system for detecting and tracking moving objects in a stationary camera visual surveillance setting has been presented. Moving objects are detected using classical background subtraction methods. The appearance of the moving objects is modeled using a mixture of mixtures of Gaussians, rather than a simple mixture of Gaussians. This appearance model have a two main advantages. Firstly, no a priori setting of mixture parameters (e.g. number of mixture components, initial means, etc.) is needed. Secondly, the computational

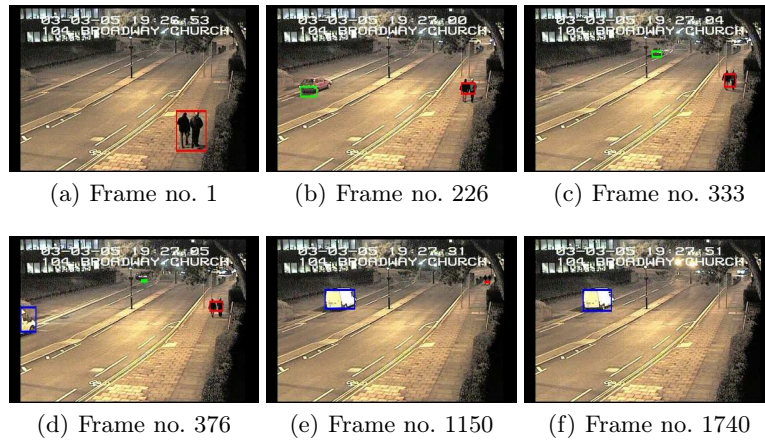


Fig. 2. People and vehicle tracking. Detection is performed concurrently with object tracking.

complexity for computing appearance likelihoods is $\mathcal{O}(n)$, which is important to achieving real-time tracking. Object tracking is performed using a particle filter framework.

Results on daylight color video sequences as well as night video sequences are presented in the paper. The results show a very robust performance with respect to scale changes and lighting conditions.

References

1. Abd-Elmaged, W., Davis, L.: Density estimation using mixture of mixtures of gaussians. In: 9th European Conference on Computer Vision. (2006)
2. Comaniciu, D., Meer, P.: Mean Shift: A Robust Approach Toward Feature Space Analysis. *IEEE Trans. Pattern Analysis and Machine Intelligence* **24** (2002)
3. Han, H., and, D.C., Zhu, Y., Davis, L.: Incremental density approximation and kernel-based baesian filtering for object tracking. In: *IEEE International Conference on Computer Vision and Pattern Recognition*. (2004)

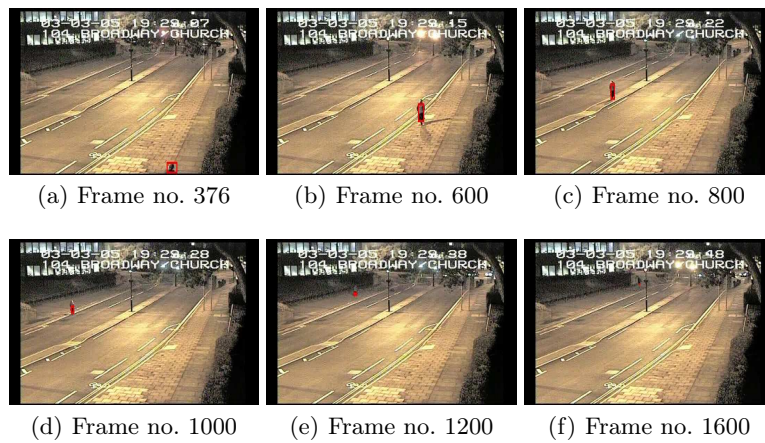


Fig. 3. People tracking under severe background ambiguity for long time periods.

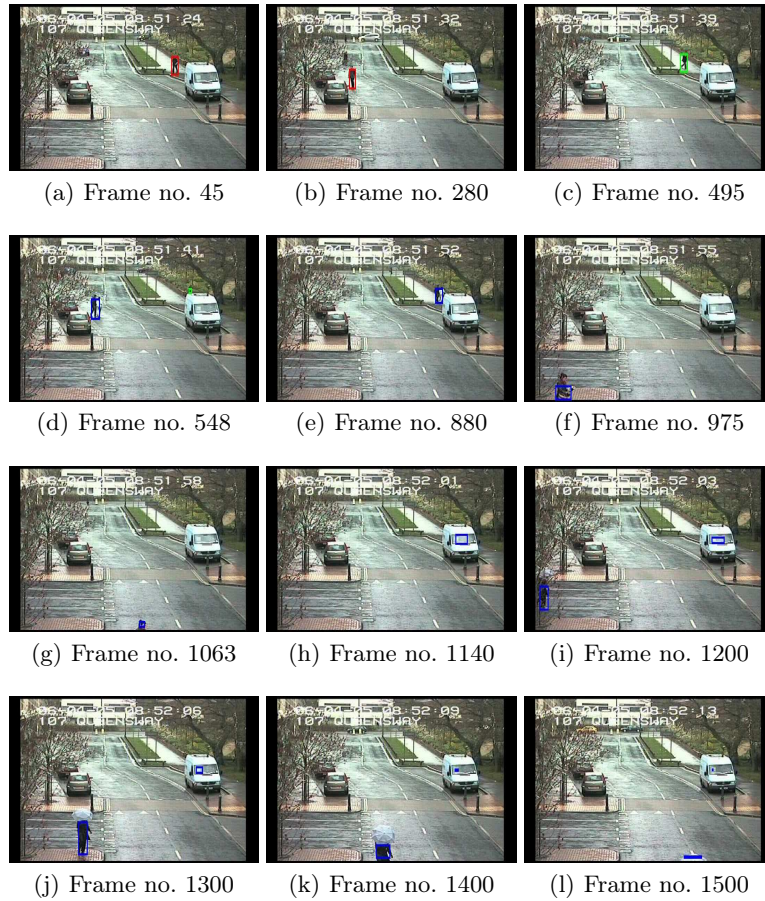


Fig. 4. People tracking on daylight color data.