An automatic traffic surveillance system for tracking and classifying vehicles
Shih-Hao Yu (游世豪), Yung-Sheng Chen (陳永盛), Jun-Wei Hsieh (謝君偉), and Wen-Fong Hu (胡文豐)
Department of Electrical Engineering, Yuan Ze University
Chung-Li 320, Taiwan, R.O.C.
Tel: (03) 463-8800 ext. 409, Fax: (03) 463-9355, e-mail: eeyeschen@ee.yzu.edu.tw

Abstract
This paper presents an automatic traffic surveillance system for tracking and classification vehicles in traffic video sequences. In order to detect moving objects from a dynamic background scene, which may have temporal clutters, we devised an adaptive background update method and a motion classification rule. A two-dimensional token-based tracking system using a Kalman filter is designed to track individual vehicles. Besides, we propose a novel method to classify vehicles by means of direct percept on visual data.

Keywords: traffic surveillance, vehicle tracking, vehicle classification.

1. Introduction
In recent years, the development of complex surveillance systems has captured the interest of both the research and industrial worlds. Strong and challenging requirements of modern society are involved in this problem, which aims to increase safety and security in several application domains such as transport, tourism, home and bank security, military applications, etc. At the same time, fast improvements in microelectronics, telecommunications, and computer science make it necessary to consider new perspectives in this field. In the past few years, traffic congestion went from bad to worse. Many people waste a lot of time and money. It results in consumption of the society resources. Many countries such as America, Japan, and French, budget millions dollars and combine another advanced technique like communication, control, information to establish a complete automatic surveillance system. It is also called intelligent transport system (ITS). That can solve the problem of traffic congestion.

In this paper, we propose a novel integrated method to classify vehicles. In background update, we utilize the information of motion to filter noise effectively. This method doesn’t update the background of moving vehicles. In objects tracking, we adopt Kalman filter to predict location of the vehicles in the next time state. And in the vehicles classification, we use fuzzy C-Means to classify vehicles. On the middle-horizontal region of anyone frame, we find that the size of vehicles is a very obvious feature. Hence we use Fuzzy C-Mean by size feature to classify vehicles first. Then, due to angle of view and vehicles structure, we utilize straight-line fitting to classify vehicles as buss and trucks. Because of precision improvement, analyzing results of different time state can obtain maximum likelihood estimation. Finally, system will compute traffic flow per unit time.

The rest of this paper is organized as follows. In the next section, we will first describe object segmentation and object tracking.
Then, the details of vehicles classification are described in section 3. Section 4 reports experimental results. Finally, a conclusion will be presented in section 5.

2. Object segmentation and tracking

2.1. Object segmentation

We compute the difference \( D_k(x, y) \) between the input images \( I_k(x, y) \) and the background image \( B_k(x, y) \):

\[
D_k(x, y) = |I_k(x, y) - B_k(x, y)| \quad (1)
\]

Then, we can establish binarized image \( T_k(x, y) \) by checking whether each point \( (x, y) \in D_k(x, y) \) is a background point or a moving-object point:

\[
T_k(x, y) = \begin{cases} 
1, & \text{if } D_k(x, y) > Th \\
0, & \text{otherwise}
\end{cases} \quad (2)
\]

After labeling, we use geometric features such as length, width, and area to filter out noise. Fig. 1 shows an example of object segmentation.

![Object segmentation results](image)

For the adaptive background, environmental changes may, however, cause a pixel's intensity to vary over time. We can use the temporal median operation \(^1\).

For frame \( I_k \) in the video sequence, the background \( B_k \) is updated as follows:

\[
B_{k+1}(x, y) = \begin{cases} 
B_k(x, y) + 1, & \text{if } B_k(x, y) < I_k(x, y) \\
B_k(x, y) - 1, & \text{otherwise}
\end{cases} \quad (3)
\]

Background \( B_k \) is updated as above unless there is a moving object in the frame \( I_k \).

2.2. Object tracking

An object tracking algorithm is used to track the detected objects by predicting their locations based on a linear prediction model, Kalman filter \(^2\).

The Kalman filter estimates a process by using a form of feedback control: the filter estimates the process state at some time and then obtains feedback in the form of measurements. As such, the equations for the Kalman filter fall into two groups: time update equations and measurement update equations. The time update equations are responsible for projecting forward the current state and error covariance estimates to obtain the a priori estimates for the next step. The measurement update equations are responsible for the feedback, i.e., for incorporating a new measurement into the a priori estimate to obtain an improved a posteriori estimate.

The time update equations are presented:

\[
\hat{x}_k = A\hat{x}_{k-1} \quad (4) \\
P_k = AP_{k-1}A^T + Q \quad (5)
\]

Where \( A \) is state transition matrix, \( x_k \) is system state at time \( k \), \( P_k \) is estimate error covariance. The measurement update equations are presented:

\[
K_k = P_kH^T\left(HP_kH^T + R\right)^{-1} \quad (6) \\
\hat{x}_k = \hat{x}_k + K_k(z_k - H\hat{x}_k) \quad (7) \\
P_k = (I - K_kH)P_k \quad (8)
\]

Where \( K_k \) is Kalman gain, \( H \) is observation.
matrix, and \( z_i \) is set of measurements.

We can track the moving object efficiently by predicting the next center coordinate from the observed coordinate of the moving object.

We apply Kalman filter on the system state \( x(k) \), which is defined as a 4-D vector of the positional change of the target object per unit time interval and the size change of the target object.

\[
x(k) = \begin{pmatrix} \Delta x_{\text{center}}(k) \\ \Delta y_{\text{center}}(k) \\ \Delta x_{\text{size}}(k) \\ \Delta y_{\text{size}}(k) \end{pmatrix}
\]  

(9)

We apply a recursive Kalman filtering operation to obtain optimal linear minimum variance. But we must use this location of prediction to find the real location. We use matching function:

\[
f_{R_i,R_j}(x_i, y_i) = \sqrt{w_x(x_i - x_j)^2 + (y_i - y_j)^2} + \sqrt{w_y(x_i - x_j)^2 + (y_i - y_j)^2}
\]  

(10)

Where \( (R_i, R_j) \) is defined as an existed vehicle and one region at this time individually. The suffix \( c \) and \( s \) are defined as center and size individually. \( w_x \) and \( w_y \) are center and size weight individually.

### 3. Vehicles classification

In the intelligent transport system, besides detection and tracking object, computing traffic flow is also a very important task.

The size of the vehicles is the most obvious feature between big vehicles and normal vehicles. Hence we use the size of vehicles appearing in the middle-horizontal region as the data feature of fuzzy C-Means.

The algorithm of fuzzy C-Means is described as follows:

- **Step 1.** Determine the number of clusters \( K > 0 \), maximum process times \( t_{\text{max}} \), a small tolerance error \( \varepsilon > 0 \).
- **Step 2.** Determine the initial center position of clusters \( c_j(0) \) for \( j = 1, \ldots, K \).
- **Step 3.** For \( t = 1, \ldots, t_{\text{max}} \)

(A) For \( i = 1, \ldots, N \)

(i) Compute the distance:

\[
d'_i = \| x_i - c_j \|, \quad j = 1, \ldots, K
\]  

(11)

(ii) Determine which cluster the data \( x_i \) belongs to:

\[
u_i = \frac{(y_i + d_i)^m}{\sum_{j=1}^{K} \left( (y_i + d_i)^m \right)}, m = 1 \text{ or } 2
\]  

(12)

(B) Update the center of cluster:

\[
c_j = \sum_{i=1}^{N} u_i^m x_i / \sum_{i=1}^{N} u_i^m
\]  

(13)

(C) Repeat step 3 until

\[E(t) = \sum_{j=1}^{K} \| c_j - c_j^{t-1} \|^2 < \varepsilon \]  

(14)

On the other hand, due to angle of view and vehicles structure, there is a very clear difference between buses and trucks. The upper edge of the buses approaches a straight-line, whereas the upper edge of the tracks doesn’t. Hence we utilize straight-line fitting to classify them. We consider the problem of fitting a set of \( N \) data points \( (x_i, y_i) \) to a straight-line model:

\[y(x) = y(x; a, b) = mx + b
\]  

(15)

We define total error function:

\[E(b,m) = \sum_{i=1}^{N} \left( \frac{y_i - b - mx_i}{\sigma_i} \right)^2
\]  

(16)

We can obtain the parameters of the straight-line fitting through derivatives of \( E(b,m) \) with respect to \( m, b \) vanish. After that, we can judge the similarity by definition of average error function:
\[ E_{xy}(b,m) = \sum_{i=1}^{N} \left( y_i - \frac{mx_i - b}{N} \right)^2 \]  

(17)

The flowchart of buses and trucks classification is shown in Fig. 2. If \( E_{xy}(b,m) \) is smaller than 2.0, this vehicle is classified as buses; on the contrary, it is classified as trucks.

![Flowchart of buses and trucks classification](image)

Although vehicles can be classified by straight-line fitting, sometimes they may be classified failure due to noise. Hence we apply maximum likelihood estimation to improve the success rate of classification.

The accumulative probability \( f_i(x|v_j) \) of input vehicle \( x \) which belongs to one kind of vehicles \( v_j \) until time state \( k \) is defined as:

\[ f_i(x|v_j) = \sum_{i=1}^{k} P_i(x|v_j) \]  

(18)

Where \( P_i(x|v_j) \) is the probability of input vehicle \( x \) which belongs to one kind of vehicles \( v_j \) at time state \( k \).

4. Experimental results

Fig. 3 is the results after tracking in the highway video sequence. Fig. 4 is the results after classification. Every vehicle is labeled its result of classification in its right side of boundary. Cars are labeled as red plus, big vehicles are labeled as blue cross, buses are labeled as blue triangle, and trucks are labeled as blue invert triangle. In order to stress results of classification,

Fig. 5 shows the results of classification at different time.

We use video sequences captured in the highway to be our test sequences. Every sequence lasts one to two minutes. Table 1 and Table 2 display statistics of tracking and statistics of classification individually.

![Results after tracking](image)
5. Conclusions

In this paper, we present an integrated method to classify vehicles. In adaptive background, first we use the information of the vehicles trajectory and time median operation to filter out noises. Then we design a tracking system using Kalman filter to track individual vehicle. Due to perspective, size is the most obvious feature to classify vehicles coarsely. Hence we use fuzzy C-Means to classify them as normal vehicles and big vehicles. On the other hand, buses and tracks can be distinguished by straight-line fitting due to angle of view.

Our success rate of classification is about 80%. Therefore it still has space to promote its success. We will add another information to promote the accuracy of our system further. Even through our classification process is very fast, our system is still processed offline. It may be a good work to develop an online system in the future.

Table 2 Statistics of classification.

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Counts of vehicles in reality</th>
<th>Counts of classifying vehicles correctly</th>
<th>Average success rate of classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Video Sequence 1</td>
<td>55</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>Video Sequence 2</td>
<td>52</td>
<td>5</td>
<td>14</td>
</tr>
<tr>
<td>Video Sequence 3</td>
<td>47</td>
<td>5</td>
<td>24</td>
</tr>
</tbody>
</table>

References


