Multiple Vehicle Detection and Tracking in Highway Traffic Surveillance Video Based on SIFT Feature Matching

Kenan Mu*, Fei Hui*, and Xiangmo Zhao*

Abstract
This paper presents a complete method for vehicle detection and tracking in a fixed setting based on computer vision. Vehicle detection is performed based on Scale Invariant Feature Transform (SIFT) feature matching. With SIFT feature detection and matching, the geometrical relations between the two images is estimated. Then, the previous image is aligned with the current image so that moving vehicles can be detected by analyzing the difference image of the two aligned images. Vehicle tracking is also performed based on SIFT feature matching. For the decreasing of time consumption and maintaining higher tracking accuracy, the detected candidate vehicle in the current image is matched with the vehicle sample in the tracking sample set, which contains all of the detected vehicles in previous images. Most remarkably, the management of vehicle entries and exits is realized based on SIFT feature matching with an efficient update mechanism of the tracking sample set. This entire method is proposed for highway traffic environment where there are no non-automotive vehicles or pedestrians, as these would interfere with the results.

Keywords

1. Introduction

Nowadays, highway traffic video surveillance systems have been receiving increasing interest from both the commercial and scientific communities. More and more work has been devoted to the research on techniques for vehicle detection and tracking in highway scenarios based on computer vision.

For a fixed camera setting, a stable background is obtained in ideal conditions. Nonetheless, if the environment is non-stationary due to things such as weather changes, variations in illumination, strong wind, earthquake and other random disturbances, then, the camera will dither. Many methods focus on building a stable background model and an optimal background update mechanism [1,2]. However, the high computational cost leads to non-real-time results for such methods, as in [3], and it is sensitive to variations in illumination. Some other methods, such as ones based on vehicle structure features (e.g., symmetry, edges, etc.) [4,5], detect vehicles without background model. However, those methods are also sensitive to various environmental conditions (e.g., weather, time of the day) and changes to the...
vehicle’s appearance (e.g., moving posture, scale changes). Therefore, detecting vehicles by using the features that are more stable against various environmental conditions is still an open issue.

As for vehicle tracking, statistical approaches, such as particle filters-based and Kalman filters-based methods [6,7], have been adopted in many recent works. However, these tracking approaches carry the risk of drifting away from the correct targets [8]. Additionally, they are often limited to a constant number of objects. Some works, such as [9], can handle the tracking of a variable number of objects, but this often requires an external method to manage where an object enters and exits from. The vehicle tracking method based on feature matching is another widely used method [10,11]. The accuracy of the feature based tracking method depends on fewer yet better features. Too many features lead to a high computational cost, while too few features lead to the decrease of tracking accuracy. Additionally, good features are unaffected by the various environmental conditions, which is important for maintaining higher tracking accuracy.

In this work, a complete method for vehicle detection and tracking that focuses on the above-mentioned limitations is proposed for the highway traffic environment. The flowchart of our proposed method is shown in Fig. 1. Our method involves a two-stage procedure, which is as follows. Firstly, vehicles are detected from the difference between two images based on feature matching. Different from a classical background subtraction-based method and frame difference-based method, our proposed method does not require an initial background image, and the difference image is obtained from the two aligned images [12].

---

**Fig. 1.** The flowchart of the vehicle detection and tracking method based on SIFT feature matching.
Secondly, through detecting and matching the SIFT features of the two images, the homography is available, and the previous image is aligned with the current image. We used the SIFT feature because it is stable against rotation, scale, changes in illumination, and noise. As such, the proposed detected method based on the SIFT feature is invariant for variations in traffic scenes. The difference between aligned images is only significant at those regions of the image features motion, that is, the regions where vehicles are. Therefore, the positions of the vehicles are obtained from the difference image.

Next, vehicle tracking is achieved by matching features. In order to decrease the time consumption of feature matching and target searching while maintaining high tracking accuracy, we only carried out matching in the candidate vehicle regions instead of the entire image. Once a candidate vehicle is detected in the current image, it is to be matched with the sample vehicles in the tracking sample set. The candidate vehicle is only to be tracked if there is a tracking sample matched with it. Additionally, the management of a vehicle’s entries and exits is intrinsically realized by the proposed update mechanism of the tracking sample set, thereby avoiding the need for an external control module.

2. Vehicle Detection

The flowchart of our proposed vehicle detection method is shown in Fig. 2. In order for the two adjacent frames to be detected, the features are detected by the Scale Invariant Feature Transform (SIFT) algorithm in the first step. In the second step, point correspondences are obtained by the K-D Tree N_N (nearest neighbors) algorithm. In the third step, the Random Sample Consensus (RANSAC) algorithm is used to eliminate false point correspondences and compute the homography matrix, which defines the geometric transformation between the two source images. In the fourth step, the current frame is transformed to be aligned with the previous frame, and the interference caused by changes in illumination and camera dithering is eliminated at the same time. In the last step, the transformed frame is differed with the previous frame to obtain the difference image, and then regarding the region with the higher Sum of Absolute Differences (SAD) as the moving vehicle.

![Fig. 2. The flowchart of the vehicle detection based on SIFT feature matching.](Image)

2.1 SIFT Feature Extraction and Matching

The successive frames collected by a fixed camera have the same background in an ideal situation, but there are differences caused by variations in illumination, camera dither, scene swing (i.e., trees swaying in the wind), and moving objects in general. Computing the geometric transformation matrix between the adjacent frames can eliminate the difference caused by the former two, and the parameters of the
matrix are determined by the point correspondences of the adjacent frames. In this paper, we use a SIFT feature detector to extract features from frames. The SIFT feature detection process consists of the steps listed below [13].

Step 1. Extreme detection in a multi-scale space: In a multi-scale space, the interest points with scale and rotation invariances are detected by the Gaussian differential function.

Step 2. Key points location: According to the position of the points of interest to locate the key points.

Step 3. Direction determination: Assign the direction for each key point according to the gradient direction of its neighbor points.

Step 4. Key points description: Compute the gradient in the neighborhood of each key point to generate a feature descriptor.

After the feature detection, we used the k-d tree N-N search algorithm to match the features of the adjacent frame images, where the Euclidean distance was regarded as the similarity criterion [14]. Fig. 3 shows an example of the SIFT feature matching of two frames.

![Image](image-url)

**Fig. 3.** SIFT feature matching: (a) the frame at time $k$, (b) the frame at time $k+5$, (c) the result of feature matching of the two frames.

### 2.2 Image Alignment and Vehicle Detection

After obtaining the point correspondences described in Section 2.1, we can compute the parameters of the homography matrix. Considering the possible variations in illumination as well as the changes in rotation, translation, and zoom caused by camera dithering, we used the projection transformation model to describe the background differences between the adjacent frames. Assume that homography matrix $H$ is as follows:

$$H = \begin{bmatrix} m_0 & m_1 & m_2 \\ m_3 & m_4 & m_5 \\ m_6 & m_7 & 1 \end{bmatrix}$$  \hspace{1cm} (1)

We used the RANSAC algorithm to estimate $H$. In the RANSAC algorithm, repeatedly selecting a random subset in the feature point correspondence set until as much as possible point correspondences satisfying the matrix $H$, which are calculated by the current subset. The current matrix $H$ is considered to be the optimal homography matrix. All of the point correspondences that satisfy matrix $H$ are the optimal matching points, while the rest of the correspondence points are rejected as mismatching [13].

Once a reliable estimation of the homography $H$ is obtained from using the RANSAC algorithm, image alignment between the previous and current frames is performed. This is achieved by warping...
the current frame with $H$ and then determining the difference between the two aligned images to obtain the candidate vehicle region. An example of image alignment and difference between aligned images is shown in Fig. 3. The difference between the aligned images is expected to be null, except for the regions of the moving vehicle. Fig. 4(c) illustrates this difference for the previous example. As can be seem, high brightness regions indicate a significant difference appearing in the areas of moving vehicles. However, some background regions do exist.

Fig. 4. Image alignment and difference image: (a) the image at time $k$ of a traffic sequence acquired with a fixed camera, (b) the image obtained after warping the image at time $k$ with $H$, (c) the difference between aligned images.

The position of moving vehicles is extracted by computing the SADs over the difference image between the aligned images and by considering the regions with a higher SAD as vehicle regions. Assume the frame at time $k$ and $k+5$ are $f_k$ and $f_{k+5}$, respectively, the warped image of $f_{k+5}$ is $f'_{k+5}$, and the difference image is $\Delta f'$. If the difference image $\Delta f'$ is equally divided into $m \times n$ regions, the size of each region is $a \times a$. The SAD value of each region is calculated as follows:

$$SAD = \sum_{i=0}^{m} \sum_{j=0}^{n} |f_k(i, j) - f'_{k+5}(i, j)|$$

(2)

Calculate the SAD value for each small region and determine the vehicle region by comparing each SAD value with the threshold $T_{SAD}$. In Fig. 5(a), the region with high brightness is the region whose SAD is larger than $T_{SAD}$, so it is regarded as the vehicle region, as shown in Fig. 5(b). In this paper, we sorted the SAD value from big to small and took the median as the SAD threshold:

$$T_{SAD} = med \{ SAD_i \mid i = 1, 2, \ldots, m \times n \}$$

(3)

Fig. 5. Result of vehicle detection: (a) the region with the SAD higher than $T_{SAD}$, (b) the vehicle region corresponded to the region in (a).
In particular, if the area of any region with a high SAD is smaller than the area threshold $T_d$, it will be considered to be a non-vehicle region. The area threshold is set for two purposes: one is for removing the interference, such as lane, shadow, and background regions; another is for avoiding having too few SIFT feature points detected as a result in the small vehicle region, which leads to the failure of feature matching in vehicle tracking.

3. Vehicle Tracking

In this section, we introduce a method of vehicle tracking based on SIFT feature matching. We took into account the detected vehicles in previous frames as the tracking samples in order to match the SIFT features with the current detected vehicle region. If the matching rate was higher than a matching rate threshold, we considered the current vehicle region to be the same as the tracking sample, and then updated the tracking sample with the current vehicle region. Otherwise, we considered the current region to be a new vehicle (a stationary vehicle starts, or a vehicle enters into the camera’s view), and then we tracked the vehicle and added it to the tracking sample set.

3.1 Tracking Sample Set Update

The proposed tracking sample set in this paper is the set of all detected vehicles in previous frames. Assume that a detected vehicle is $v_i^k$, then the tracking sample set is represented as follows:

$$V = \{v_i^k | i \in m, k \in n\}$$  (4)

where, $m$ is the length of the sample set and $n$ is the number of the detected frames in the traffic sequence.

Updating the tracking sample set in real time can prevent the degradation of tracking sample set. There are four reasons that lead to the degradation of the tracking sample set, which correspond to the four sample set updates described below.

1. If a vehicle, whose corresponding sample existed in the tracking sample set, is detected in the current frame, then update the corresponding vehicle sample with the current detected vehicle. With the vehicle moving on the road (driving posture, close to the camera or far away from it), the shape of the same vehicle is significantly different from what is in each frame over a period of time. So, updating the corresponding vehicle sample with the current detected vehicle can efficiently avoid degradation.

2. If a new vehicle enters into the camera’s view, then add it to the tracking sample set.

3. If a vehicle leaves the camera’s view, then remove the corresponding sample from the tracking sample set.

4. If a vehicle stop moving, then remove it from the tracking sample set as the background region.

3.2 Matching Rate

In order to measure the matching degree between the vehicle to be tracked and the tracking sample, we propose the matching rate in this section. Assume $N_i$ features are detected in the tracking sample
that \( N_2 \) features are detected in the current detected vehicle region in the frame at time \( k + 1 \), and that the \( N \) point correspondences are found. Then, the matching rate, \( Rate \), is calculated as follows:

\[
Rate = \frac{N}{\max\{N_1, N_2\}}
\]  

The matching result between the tracking sample and vehicle to be tracked is shown in Fig. 6 and Fig. 6(a) represent that the \( Rate \) is larger than a matching rate threshold, while (b) represents the case where the \( Rate \) is smaller than the matching rate threshold.

3.3 Judgment of Vehicle Entries, Exits, and Stops

There are three special cases that have to be considered in vehicle tracking, namely vehicle entries, exits, and intermittent movements. We focused on these three cases and the corresponding methods for determining the judgment and solution are introduced below.

1. For a vehicle detected in the frame at time \( k \), calculate the matching rate (\( Rate \)) between the vehicle and each tracking sample in the tracking sample set. If all of the \( Rates \) are smaller than the matching rate threshold \( T_k \), then the vehicle is considered to be the one entries camera’s view and is added into the tracking sample set as a new sample.

2. For a sample in the tracking sample set, if no matching vehicle is detected in the frame at time \( k \) and \( k + 1 \), where the matching rate (\( Rate \)) between the sample and each detected vehicle in the frame at time \( k \) and \( k + 1 \) is smaller than \( T_k \), then the sample is considered to correspond to the vehicle exiting from or stopping in the camera’s view and it is removed from the tracking sample set.

3. For a sample in the tracking sample set, if its corresponding vehicle has not moved significantly in the frame at time \( k \) and \( k + 1 \), then the sample is considered to be corresponding to the vehicle stops or be the interference and is removed from the tracking sample set.

In the third case, the way to judge the moving vehicle should be proposed. The basis is that whether the offset of the corresponded detected window’s geometric center is smaller than an offset threshold. Assume the coordinates of the current vehicle’s detected window are \((x_1, y_1)\) and \((x_2, y_2)\) (the coordinates of the upper left point and lower right point, respectively), then the geometric center is calculated as follows:
The offset of the detected window’s geometric center for the same vehicle between the previous detection and current detection is calculated as follows:

\[
s_i = \sqrt{(x_{i2} - x_{i1})^2 + (y_{i2} - y_{i1})^2}
\]

The offset threshold is calculated as follows:

\[
\begin{align*}
T_s &= \alpha vt_i \\
\alpha &= \frac{X \times Y}{320 \times 240} \\
v &= \frac{1}{M} \sum_{i=1}^{M} s_i / t_2 \\
t_1 &= \frac{m}{F}, t_2 = \frac{M}{F}
\end{align*}
\]

\[\alpha\] is the normalized resolution parameter, which is the rate of the resolution of the current frame \(X \times Y\) and the standard resolution \(320 \times 240\). We can understand the expression (8) by combining it with Fig. 7, which shows the moving track of a vehicle as enters and exits the camera’s view. The detected window’s geometric center showed as a point is represented the vehicle. Assume that the frame rate of the current traffic sequence is \(F\). \(M\) frames are collected from the vehicle entering and exiting the camera’s view, and the time consumption is \(t_2\). \(v\) is the average moving velocity of the detected window’s geometric center of the vehicle. The unit for \(v\) is a pixel. \(m\) is the number of frames between the two frames to be detected in our experiment, and \(t_1\) is the corresponding time interval. Then, the offset threshold \(T_s\) can be determined according to (8), and we can judge the stopped vehicle or interference and remove it from the tracking sample set according to the criterion \(\Delta S \leq T_s\).

Fig. 7. Moving track of the detected window’s geometric center of a vehicle from entering in to exiting from camera’s view.
4. Experiments

4.1 Threshold Settings

The parameters we referred to are the \(SAD\) threshold \(T_{SAD}\), area threshold \(T_{A}\), matching rate threshold \(T_{R}\), and the offset threshold \(T_{E}\) of the detected window’s geometric center. The determination of \(T_{SAD}\) was introduced in Section 2.2, and the calculation method of \(T_{E}\) was introduced in Section 3.3. We determined the thresholds \(T_{A}\) and \(T_{R}\) through experiments. When \(T_{A} = 33\) and \(T_{R} = 80.31\%\), the best experiment result has then been obtained.

4.2 Experiments and Analysis

In this section, we detect and track vehicles in different traffic sequences by using our proposed method. A fixed roadside camera collects each traffic sequence. Overall, an average detection rate of above 87% and tracking accuracy of 90% are obtained for different traffic scenarios, including variations in illumination and traffic conditions (except for rainy, greasy, snowy, and nighttime). The system operates at a frame rate of between 5–10 fps.

Fig. 8 shows the difference images of the two frames in three traffic sequences. As can be seen, the moving vehicle region has a higher \(SAD\) than the background region. The region whose \(SAD\) is larger than \(T_{SAD}\) is regarded as the candidate vehicle region. The small background region involved in the candidate vehicle regions can be removed by comparing the area with \(T_{A}\); while the larger background region will be removed in vehicle tracking, which is demonstrated below.

\[\text{Fig. 8. Difference images of the two frames in three traffic sequence.}\]

The traffic sequence shown in Fig. 9 has interference caused by trees swaying in the background. So it is easily led to detecting the trees as vehicles, as shown in Fig. 9(a). For eliminating this large interference region, we computed the offset of the geometric center of the detected window for the same vehicle in the adjacent frames. The method is referred to in Section 3.3. If the region is a background region, the offset of the detected window’s center is small between the adjacent frames. Then, we were able to remove the interference from the tracking sample set when the offset was smaller than the area threshold. This resulted in the interference region not existing in the next frame (at time \(k+15\)), as shown in Fig. 9(c).
Fig. 9. Larger background region is removed in vehicle tracking: (a) larger background region caused by trees swinging in the frame at time $k$; (b) the same background region with a small offset in the frame at time $k + 5$; (c) the larger background region is eliminate in the frame at time $k + 15$.

Fig. 10. Detecting and tracking result with the proposed method in four traffic sequences.

Fig. 10 shows the detecting and tracking results with the proposed method in four traffic sequences. In the first sequence (first row), there are variations in illumination and shadows; in the second and third sequences (second and third rows), the camera dithers during the sequence collection; in the fourth sequence (fourth row), there is intermittent vehicle motion. Taking the first sequence as an example, the vehicle (highlighted by the green rectangle) was detected and tracked in the frames at time $k$ and $k + 5$, and it was tracked constantly in the frame at time $k + 15$. This is because the vehicle was nearly out of the camera’s view in the frame at time $k + 15$ (vehicle exits) and the rest of the structure of
the vehicle in the frame failed to match (matching rate is lower than $T_g$) with the corresponding tracking sample. The corresponding sample was then removed from the sample set. At the same time, the vehicle (highlighted by the green rectangle) was detected and began to be tracked in the frame at time $k + 15$ (vehicle entries). This is because the vehicle’s area was smaller than the area threshold $T_a$ at time $k$ and $k + 5$ until time $k + 15$. So, in the frame at time $k + 15$, the vehicle was detected and added into the tracking sample set as the entering vehicle. Taking the fourth sequence as another example, the vehicle (highlighted by the red rectangle) as not detected until time $k + 260$ (vehicle intermittently moves). This is because the vehicle was static before time $k + 260$, so it was ignored as being a part of the background before beginning to move at time $k + 260$. As such, it was detected in the frame at time $k + 260$ and added to the tracking sample set.

Table 1 shows the time consumption of each vehicle tracking based on the particle filters in [12], the SIFT feature matching points in [15], and our proposed tracking method. Taking the traffic sequence, as shown in the first row in Fig. 10, it can be observed that compared with the tracking methods used in [12,15], the average time consumption of each vehicle tracking using the proposed method was the lowest. So, our method can be used in real-time applications.

According to the result of our experiments, it can be seen that our proposed method can reliably detect and track vehicle in various traffic scenes. Indeed, compared to classical methods for vehicle detection, such as those based on vehicle structure (e.g., edges, symmetry, etc.) or specific application features (e.g., vehicle shadow), the proposed method is less affected by a particular traffic scene (e.g., variations in illumination, shadows, camera dithering, etc.). Meanwhile, the proposed method is also good for dealing with some special situation, such as the vehicle’s entry into exit from a camera’s view and intermittent movements.

Table 1. Time consumption of each vehicle tracking based on particle filters in [12], SIFT feature-points matching in [15] and proposed tracking method in this paper

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.801</td>
<td>0.813</td>
<td>0.209</td>
</tr>
<tr>
<td>B</td>
<td>0.747</td>
<td>0.709</td>
<td>0.124</td>
</tr>
<tr>
<td>C</td>
<td>0.740</td>
<td>0.776</td>
<td>0.227</td>
</tr>
<tr>
<td>D</td>
<td>0.816</td>
<td>0.747</td>
<td>0.188</td>
</tr>
</tbody>
</table>

The unit of time is the second (s).

5. Conclusions

In this paper, a complete method for vehicle detection and tracking in highway traffic surveillance video by using a roadside camera has been presented. The system first detects vehicles based on successive image rectification using plane-to-plane homography. After SIFT feature detection and matching is conducted, the RANSAC algorithm estimates the homography matrix, and then image alignment is achieved. In the difference image, we detected moving vehicles by searching for the region with a higher $SAD$. We also used SIFT feature matching to track vehicles. By matching the detected vehicles in the current frame with tracking samples, each vehicle was correctly tracked in successive
frames. Our experiments showed that the proposed method is not only able to provide robust vehicle tracking in the sequence, but also to efficiently handle entry, exit, and intermittent motion management. In regards to possessing a high detection rate, tracking accuracy, and low time consumption, our proposed method is recommended for use in traffic video surveillance systems and traffic information systems.

Acknowledgement

This work has been supported by the NSFC (51278058) and Specialized Research Fund for the Doctoral Program of Higher Education of China (20120205120002).

References


Kenan Mu

She has received her M.S. degree, and is studying for the Ph.D. in the School of Traffic Information Engineering & Control from Chang’an University. Her currently research interests including image processing and Advanced Driver Assistance System (ADAS).

Fei Hui

He received his Ph.D. in Department of Computer System Architecture from Xi’an Institute of Microelectronics Technology in 2009. He has worked in Chang’an University since 2009, and he has been involved in the China “863” project (as a technical director), information technology major project of Transportation Ministry (as a technical director), National Natural Science Foundation, etc. His current research interests include embedded image processing technology and on-board embedded technology.

Xiangmo Zhao

He received his Ph.D. from Chang’an University. He is a vice president, director of the Institute of Computer Application, and leader of the national key subjects-Traffic Information Engineering & Control in Chang’an University. He is currently a director of the Information Professional Committee and member of Advisory Expert Group of China Transportation Association, a member of the National Motor Vehicle Operation Safety Testing Equipment Standardization Committee and leading group of the National Traffic Computer Application Network, vice chairman of the Institute of Highway Association on Computer Professional Committee, deputy director of the Institute of Computer in Shaanxi Province, etc. In addition, he is the editorial board member of Journal of Traffic and Transportation Engineering, Chinese Journal of Highway, ICIC Express Letters Part B: Applications, etc. He won a second prize of the National Scientific and Technological Progress Award, 3 first prize of Shaanxi Science and Technology Award, a first prize of Shaanxi Traffic Science and Technology Progress Award, etc. He is the author of more than 120 publications and holds more than 25 granted patents.