

# Vehicle Tracking for Urban Surveillance

William Roberts, Leslie Watkins, Dapeng Wu, and Jian Li

Department of Electrical and Computer Engineering

University of Florida Gainesville, FL 32611

Correspondence author: Prof. Dapeng Wu,

wu@ece.ufl.edu,

<http://www.wu.ece.ufl.edu>

## ABSTRACT

Tracking is widely used in a variety of computer vision applications, ranging from video surveillance to medical imaging. The principal goal of tracking is to first identify regions of interest in a scene, and to then monitor the movements or changes of the object through the image sequence. In this paper, we focus on unsupervised vehicle tracking for low resolution aerial images taken from an urban area. Various optical effects have traditionally made this tracking problem very challenging. Objects are often lost in tracking due to intensity changes that result from shadowed or partially occluded regions of an image. Additionally, the presence of multiple vehicles in a scene can lead to mistakes in tracking and significantly increased computation time. We propose a feature-based tracking algorithm herein that will seek to mitigate these limitations. To first isolate vehicles in the initial frame, we apply three-frame change detection to the registered images. Feature points are identified in the labelled regions using the Harris corner criteria. To track a feature point from one frame to the next, we search for the point around a predicted location, determined from the feature's previous motion, that minimizes the sum-of-squared-differences value. Finally, during the course of the image sequence, our algorithm constantly searches for new objects that might have entered the scene. We will demonstrate the success of our tracking approach through experimental considerations.

**Keywords:** Feature-based tracking, KLT tracker, Harris criteria, Sum-of-squared differences(SSD)

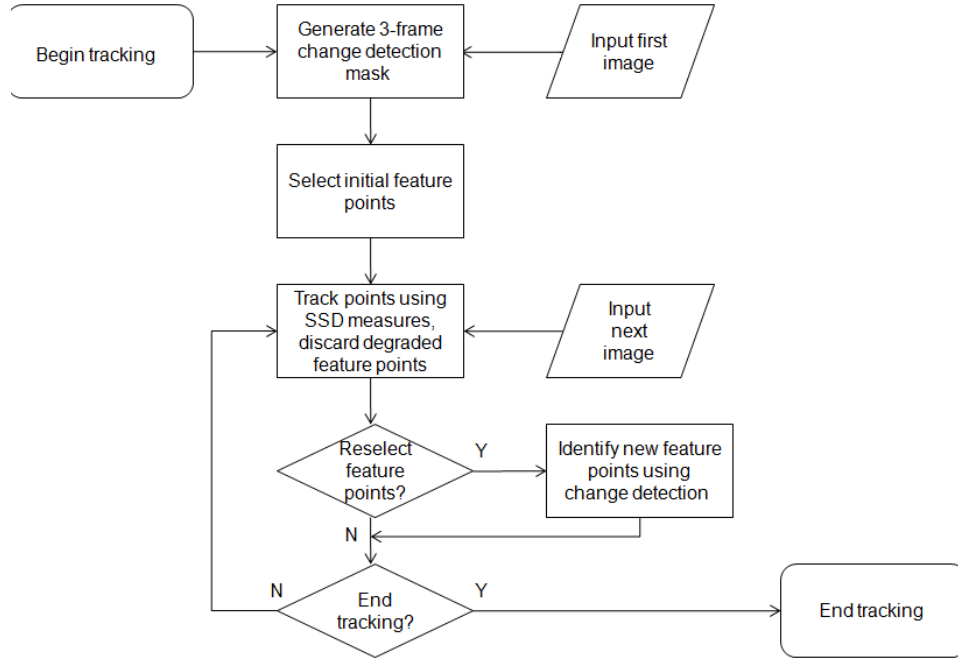
## 1. INTRODUCTION

Object tracking is a critical component in many computer vision applications, including medical imaging, surveillance, and robotics. Feature-based tracking, specifically, refers to the detection and monitoring of points, ideally located on objects of interest, through an image sequence.

The first step in any feature tracking algorithm is to identify a set of interesting points in an initial image. Tracking is then performed in subsequent frames by establishing correspondence between the current set of feature points and candidate points in the next image. Correspondence can be measured through various techniques, including correlation-based methods and sum-of-squared difference (SSD) determinations. For larger images, however, ambiguity often arises when multiple points display similar properties to the feature point of interest.

To prevent this ambiguity, and furthermore avoid the excessive time required to perform an exhaustive search, Lucas and Kanade presented an algorithm in Ref. 1 that instead uses the spatial intensity gradient of the image to direct the search for feature points. By attaining correspondence of features, they were able to register a set of images related by any linear transformation. In Ref. 2, Tomasi and Kanade further generalized this algorithm to include feature-based tracking. For their approach, points were identified as features if they possessed good properties for tracking. During the tracking process, their algorithm monitors the residue of the feature, which refers to the point's change in appearance from the initial image to the current one. If the feature's residue becomes too large, it is declared no longer suitable for tracking. In this way, they again avoid potential errors in correspondence. However, in a realistic image sequence, the number of successfully tracked points would almost certainly decrease, as points could leave the scene or perhaps become affected by occlusion or lighting variations.

As an alternative to computing the image gradient, a Kalman filter<sup>3,4</sup> could instead be used to direct the search for feature points. This approach uses knowledge of the point's motion through previous frames to predict



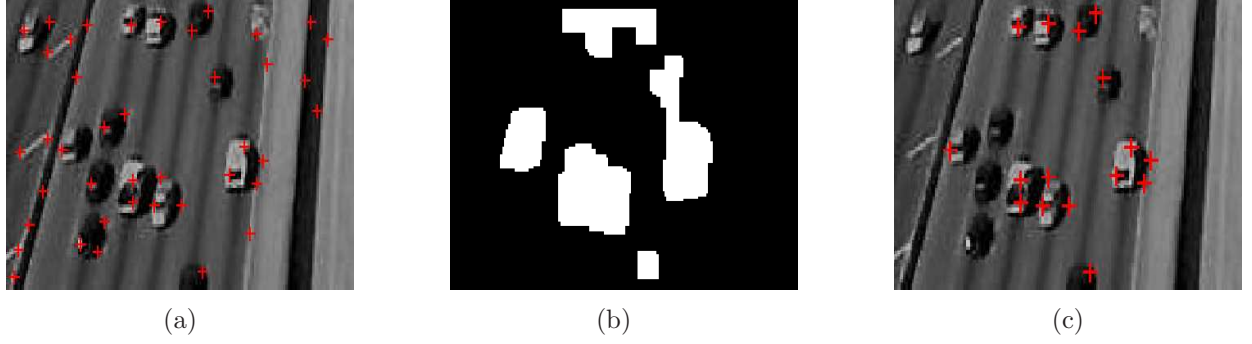
**Figure 1.** Algorithm flowchart.

where the point should lie in the current frame. After using the Kalman filter to smooth the object's trajectory, researchers often then use SSD measures to look for the feature point's best match in the directed area.<sup>5,6</sup> A Kalman filter can also provide further benefit to a tracking algorithm. In Ref. 7, researchers use the Kalman filter to design an optimal template of the marked object, and thus help to detect possible occlusions that might occur in the video frames.

In this paper, we will investigate the application of object tracking to vehicle and traffic surveillance. Several different approaches to this problem have already been presented by researchers. In Ref. 8, a feature-based tracking algorithm is described that uses a Kalman filter to direct the search for features. Detected feature points are then clustered together to identify vehicles in the scene. This approach requires the researcher to identify fiducial, or fixed, points in the images, and furthermore demands high resolution images so that multiple points on an object can be marked. In contrast, researchers in Ref. 9 and Ref. 10 attempt to segment vehicles from the video sequence by creating a model for the image background. When significant motion is present in the scene, however, this type of segmentation approach would demand excess time in order to constantly update the background model. A similar approach is taken in Ref. 11, where objects are segmented using a block matching technique. However, this algorithm requires knowledge of the pixels' intensity changes over time, and demands a large amount of frames, and thus time, to create an accurate background model.

We will present herein a new approach to vehicle tracking that will serve to overcome the deficiencies in previous algorithms. We will adopt a feature-based tracking method that will specifically address the following problematic scenarios:

- A vehicle changes direction.
- Two vehicles pass each other.
- A large number of vehicles are present in the scene.



**Figure 2.** (a) Image with marked feature points. (b) Binary object mask. (c) Image with feature points after application of mask.

- A vehicle becomes temporarily occluded.

The algorithm will be capable of tracking vehicles even for low resolution images, and will also seek to minimize computation time, a necessary demand in many realistic applications.

The remainder of this paper is organized as follows. In Section 2, we describe our approach to feature-based vehicle tracking. In Section 3, the performance of the algorithm is tested on various situational video sequences. Finally, conclusions and future considerations are offered in Section 4.

## 2. TRACKING ALGORITHM

In this section, we will describe the different components of our feature-based tracking algorithm. The algorithm is summarized in the flowchart shown in Fig. 1.

### 2.1. Feature Selection

The first step in any feature-based tracking algorithm is to identify a set of candidate points in the initial frame of a video sequence. We measure the quality of a point using the Harris corner criteria, which is essentially a measure of the intensity variation in a neighborhood around the pixel. If the relative coordinates of the pixel are given by  $\mathbf{x} = [x, y]^T$ , then the quality of the point is measured by

$$C(\mathbf{x}) = \det(\mathbf{G}) + k \times (\text{trace}(\mathbf{G}))^2, \quad (1)$$

where  $\mathbf{G}$  is a  $2 \times 2$  matrix given by

$$\mathbf{G} = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix}, \quad (2)$$

with  $I_x$  and  $I_y$  representing the horizontal and vertical intensity gradient, respectively, at the pixel of interest, and  $k$  is a scalar parameter determined by the user.

If the corner criteria exceeds a chosen threshold, then the point is selected as a chosen point for tracking. In image regions that contain a significant amount of intensity change, we further require a sufficient separation between tracking points so that the candidate feature list is sufficiently distributed in the image.

After collecting a set of predominant feature points in an image, we then attempt to remove those points that are contained in the background and thus only retain those pixels that lie on vehicles in the frame. To isolate the feature points of interest, we use three-frame change detection, or image differencing, to generate a binary mask. Thus, points will be selected from only those regions in the image where motion has occurred in the previous three frames. We further smooth and denoise the mask using a basic set of morphological filters, whose parameters are determined from the average size of vehicles in the image sequence. The feature selection process is illustrated in Fig. 2.



**Figure 3.** Image 50 in highway scene containing multiple vehicles.

## 2.2. Feature Tracking

After obtaining a set of feature points in the initial frame, we then wish to track the movement of those points over subsequent frames. To track a single point feature, an algorithm must identify a region of possible matches with which to search, and furthermore adopt a criteria to measure the quality of each possible match.

In our approach, we will use the previous motion of the feature point to predict where the point will lie in later images. For the second image, no motion history is available to the algorithm. Thus, a somewhat large search space will be defined around the target’s previous location, as the movement of the feature could occur in any direction. After a match is found in the second frame, a motion vector is then defined by differencing the location of the feature in the first frame to its location in the second frame. In this way, the location of the feature in the third image is estimated, so that only a smaller, refined search space is needed. This process is continued for all successive images. By shrinking the search space, computation time and possibilities of a false match are both reduced. Although the movement of the feature point might change from one frame to the next (for example, as a vehicle changes direction), we assume that the frame rate is sufficiently high so that the point’s motion is approximately constant for temporally adjacent images.

To determine the similarity between a feature point and a possible match, we use the sum-of-squared-differences (SSD) criteria. If a feature point on an image occurring at time  $t$  is located at  $\mathbf{x} = [x, y]^T$ , then the intensity values of a neighborhood of pixels centered at  $\mathbf{x}$  is given by  $\mathbf{I}(\mathbf{x}, t)$ . The best match for the feature point in an image taken at time  $t + \delta t$  is the point located at  $\mathbf{x} + \Delta$ , where  $\Delta = [\delta x, \delta y]^T$  is the displacement vector that minimizes the SSD measure:

$$E(\delta x, \delta y, t) = [\mathbf{I}(\mathbf{x} + \Delta, t + \delta t) - \mathbf{I}(\mathbf{x}, t)]^2. \quad (3)$$

If the SSD value of the best possible match for a feature point is still greater than some user-defined threshold, then the feature is declared no longer suitable for tracking.

## 2.3. Feature Reselection

Various types of occlusion can cause features to be temporarily lost throughout the course of a track. We have implemented a feature reselection option that allows us to search for and recover features that may have been lost. Further, by periodically searching for new feature points, we are able to detect objects that were not present in the initial scene, or perhaps were not moving in the initial scene (as image differencing is used to generate the detection mask).

Feature reselect uses the same principles as the methods that are used to identify features in the initial frame. After user-determined intervals, we calculate the number of features that have been lost throughout that interval. We then attempt to recover as many features as we lost by creating a mask that makes points of interest more prevalent, then applying Harris corner detector. However, the original quality thresholds still hold, meaning that not all features must be recovered during the reselection process. This accommodates points that have moved out of the frame, which is common when tracking moving features.



(a)



(b)



(c)

**Figure 4.** Turning vehicle sequence: (a) Frame 1. (b) Frame 11. (c) Frame 21.

### 3. NUMERICAL RESULTS

In this section, we present several examples to demonstrate the success of our algorithm. Although our approach is mostly unsupervised, there are a few variables that must be set by the user. All parameters are dependent on the proximity of the camera to the scene, and the resolution of the camera.

As mentioned, a point is identified as a feature based on its quality, as measured in Equation (1). For each video sequence, a window size (in pixels) of  $5 \times 5$  was used to determine a point's corner criteria. In contrast, a  $7 \times 7$  window was used for all SSD measurements in the tracking phase. To prevent erroneous results around the edges of image frames, an appropriately sized boundary was marked out of consideration. Finally, the researcher also chose appropriate thresholds for feature selection and feature tracking, which was determined, again, based on the size of vehicles in the video.

#### 3.1. Multiple vehicles

We first consider a video containing multiple vehicles moving in the scene, as we hope to demonstrate that the algorithm is not limited by the number of cars present. The video is captured from just above a highway, and shows vehicles travelling in a linear path across the images. Vehicles continuously enter and leave the scene, and up to 12 cars are present at one time.

Frame 50 in the 75 frame video sequence is shown in Fig. 3, with feature points denoted by the red plus signs. For this image, 8 out of 9 vehicles in the scene are identified by feature points; this result was similar for other frames in the sequence. The algorithm is able to track almost all of the cars, including those that enter the scene during the video. However, as in the image shown, the algorithm has some difficulty tracking the darker, slower moving vehicles, which only start moving about halfway through the video.

### 3.2. Turning vehicle

In the next example, we consider the ability of our algorithm to track a car as it changes direction. Specifically, we use a traffic video containing a vehicle that makes a right turn. For this sequence, we have applied an additional mask to isolate the vehicle of interest and remove uninvolved feature points. Furthermore, we do not apply feature reselection, but only select features in the initial frame.

In Fig. 4, we show frames 1, 11, and 21 of the video sequence, which contain the vehicle before, during, and after the turn, respectively. As shown, the car is successfully tracked throughout the maneuver. We also applied the KLT algorithm to this video. However, all feature points were lost as soon as the vehicle began to change direction. All the tracking parameters were held constant for both algorithms.

### 3.3. Passing vehicles

The third video sequence contains two vehicles that pass each other travelling in opposite directions on a highway. We thus hope to show that the algorithm is able to track multiple vehicles, independent of their proximity and direction of motion. As in the previous example, we have applied a mask to this sequence isolating the cars of interest. To demonstrate the effectiveness of our approach, we do not apply feature reselection, although this technique would certainly further enhance our results.

Frame 1, 6, and 11 of the video sequence are shown in Fig. 5, which show the cars before, during, and after they pass each other. Both cars are successfully tracked throughout the video sequence. When we applied our algorithm to scenes involving more vehicles (again travelling in different directions relative to each other), the results were consistent with this example. We also applied the KLT algorithm to this sequence; the results were similar to the ones obtained with our algorithm.

### 3.4. Reappearing vehicle

Finally, we highlight the effectiveness of our feature reselection process by considering a video sequence containing a vehicle that undergoes temporary occlusion. In Fig. 6, we show a vehicle before, during, and after partial occlusion by a stationary object in the scene. Several points are initially identified on the object, but all feature points on the vehicle are lost when the vehicle crosses behind a road sign. As the car reappears in the third frame, using feature reselection, features on the vehicle are recovered and it is reestablished as a point of interest.

## 4. CONCLUSION

In this paper, we have described a complete, feature-based algorithm for tracking vehicles in an urban environment. Our approach is capable of tracking vehicles in a variety of situations, and is not limited by the number of cars present in a given scene. Furthermore, the algorithm has proven computationally inexpensive.

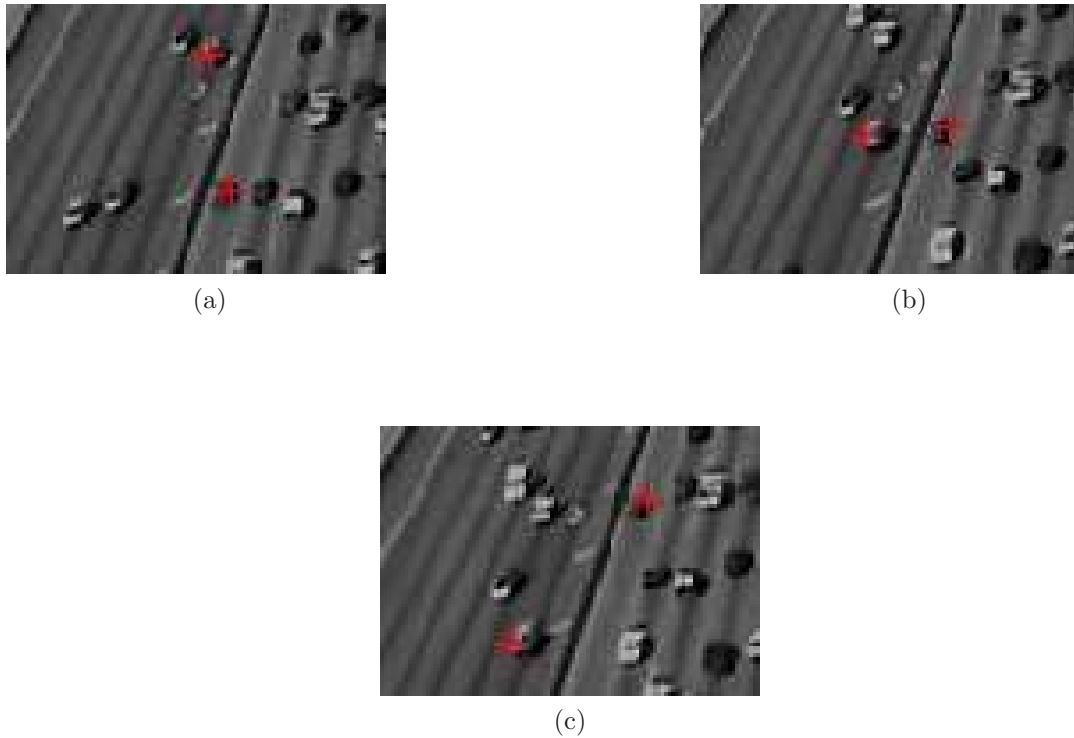
For realistic video sequences, shadowed regions often arise, which can cause the pixel intensities of objects in the shadowed areas to change significantly. Most tracking methods, including the SSD approach used herein, perceive the shadowed effect as occlusion, and would often lose a vehicle as it moved into such an area. In future considerations, we hope to address this issue, and thus further improve on the success of our approach. Additionally, we hope to incorporate the idea of memory and labelling into the algorithm. In other words, we intend to establish correspondence between feature points, so that rather than tracking points in the image, we might instead track and store an object's intensity template.

### Disclaimers

The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of AFRL or the U.S. Government.

### Acknowledgement

This material is based on research sponsored by AFRL under agreement number FA8650-06-1-1027. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon.



**Figure 5.** Passing vehicle sequence: (a) Frame 1. (b) Frame 6. (c) Frame 11.

## REFERENCES

1. B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," *Proc. of Image Understanding Workshop*, pp. 121–130, 1981.
2. C. Tomasi and T. Kanade, "Factoring image sequences into shape and motion," *Proc. of the IEEE Workshop on Visual Motion*, pp. 21–28, 1991.
3. R. E. Kalman, "A new approach to linear filtering and prediction problems," *Trans. of the ASME - Journal of Basic Engineering* **82**, pp. 35–45, 1960.
4. R. E. Kalman and R. S. Bucy, "New results in linear filtering and prediction theory," *Trans. of the ASME - Journal of Basic Engineering* **83**, pp. 95–107, 1961.
5. G. L. Jr. and T. Young, "A mathematical model for computer image tracking," *IEEE Trans. on PAMI* **4**(6), pp. 583–594, 1982.
6. N. Papanikolopoulos, P. Khosla, and T. Kanade, "Visual tracking of a moving target by a camera mounted on a robot: a combination of control and vision," *IEEE Trans. on Robotics and Automation* **9**, pp. 14–35, 1993.
7. M. Worring, H. T. Nguyen, and R. V. D. Boomgaard, "Occlusion robust adaptive template tracking," *Proc. of International Conf. on Computer Vision*, pp. 678–683, 2001.
8. B. Coifman, D. Beymer, P. McLauchlan, and J. Malik, "A real-time computer vision system for vehicle tracking and traffic surveillance," *Transportation Research: Part C* **6**(4), pp. 271–288, 1998.
9. S. Gupte, O. Masoud, R. F. K. Martin, and N. Papanikolopoulos, "Detection and classification of vehicles," *IEEE Trans. on Intell. Transport. Syst.* **3**(1), pp. 37–47, 2002.



(a)



(b)



(c)

**Figure 6.** Occluded vehicle sequence: (a) Frame 1. (b) Frame 30. (c) Frame 50.

10. S. Chen, M. Shyu, S. Peeta, and C. Zhang, "Spatiotemporal vehicle tracking: the use of unsupervised learning-based segmentation and object tracking," *IEEE Robotics and Automation Magazine* **12**(1), pp. 50–58, 2005.
11. S. Kamijo, Y. Matsushita, K. Ikeuchi, and M. Sakauchi, "Traffic monitoring and accident detection at intersections," *IEEE Trans. on Intell. Transport. Syst.* **1**(2), pp. 108–118, 2000.