A feature-based tracking algorithm for vehicles in intersections

Nicolas Saunier and Tarek Sayed
Departement of Civil Engineering, University of British Columbia
6250 Applied Science Lane, Vancouver BC V6T1Z4
{saunier,tsayed}@civil.ubc.ca

Abstract Intelligent Transportation Systems need methods to automatically monitor the road traffic, and especially track vehicles. Most research has concentrated on highways. Traffic in intersections is more variable, with multiple entrance and exit regions. This paper describes an extension to intersections of the feature-tracking algorithm described in [1]. Vehicle features are rarely tracked from their entrance in the field of view to their exit. Our algorithm can accommodate the problem caused by the disruption of feature tracks. It is evaluated on video sequences recorded on four different intersections.
Keywords: intelligent transportation systems, vehicle tracking, features, intersection

1 Introduction

Among the most important research in Intelligent Transportation Systems (ITS) is the development of systems that automatically monitor the traffic flow on the roads. Rather than being based on aggregated flow analysis, these systems should provide detailed data about each vehicle, such as its position and speed in time. These systems would be useful in reducing the workload of human operators, in improving our understanding of traffic and in alleviating such dire problems as congestion and collisions that plague the road networks [12].

Monitoring based on video sensors has a number of advantages. First, they are easy to use and install, especially when compared to magnetic loop detectors, which are the most common traffic sensor and require digging up the road surface. Second, video sensors offer the possibility to get rich description of traffic parameters and to track vehicles. Third, large areas can be covered with a small number of video sensors. Fourth, the price of image acquisition devices and powerful computers is rapidly falling. Video sensors allow to collect rich information and achieve detailed traffic analyses at the level of the vehicle.

Highways have attracted considerable attention, as they carry much of the traffic, at the expense of other parts of the road network. There is much more demand for highways, as the number of dedicated commercial systems shows it. Intersections constitute a crucial part of the road network, especially with respect to traffic safety. This is where vehicles from different origins converge. Many collisions occur in intersections, and their accurate monitoring would help understand the processes that lead to collisions and address the failures of the road system. Maurin *et al.* state in [10] that despite significant advances in traffic sensors and algorithms, modern monitoring systems cannot effectively handle busy intersections.

Figure 1: Illustration of an intersection with mixed traffic, vehicles and pedestrians, on the major and minor roads.

For all its advantages, video data is difficult to interpret. Vehicle tracking in intersections entails new problems with respect to highways, which are related to the highly variable structure of the junctions, the presence of multiple flows of the vehicles with turning movements, the mixed traffic that ranges from

pedestrians to lorries and vehicles that stop at traffic lights. Specific classification and occlusion management techniques are required. Other common problems are global illumination variations, multiple object tracking and shadow handling.

Among the many approaches to tracking in video data, the feature-tracking approach has distinct advantages, the main one being to be robust to partial occlusions. The most renowned feature-based tracking algorithm was proposed by Beymer *et al.* in [1]. It is however only applied to highway portions, with a given entrance region where vehicles are detected and a given exit region, and mostly straight complete feature tracks in-between.

This paper describes a feature-based tracking algorithm which extends the approach of [1] to intersections, with multiple entrance and exit regions, variable trajectories and possible feature track disruption. The related work and various tracking methods are presented in the next section. The approach of [1] and our adaptation is described in section 3. The results of the evaluation on multiple traffic sequences are commented in section 4.

2 Related Work

There are four main approaches for object tracking, which are sometimes combined in so-called hybrid approaches [5].

2.1 3D Model-based tracking

Model-based tracking exploits the a priori knowledge of typical objects in a given scene, e.g. cars in a traffic scene, especially with parametric three-dimensional object models [6]. These methods localize and recognize vehicles by matching a projected model to the image data. This allows the recovering of trajectories and models (hence the orientation, contrary to most other methods) with high accuracy for a small number of vehicles, and even to address the problem of partial occlusion

The most serious weakness is the reliance on detailed geometric object models. It is unrealistic to expect to be able to have detailed models for all vehicles that could be found on the roadway.

2.2 Region-based tracking

The idea in region- or blob-based tracking is to identify connected regions of the image, blobs, associated with each vehicle. Regions are often obtained through background subtraction, for which many methods exist, and then tracked over time using information provided by the entire region (motion, size, color, shape, texture, centroid). Many approaches use Kalman filters for that purpose [9, 10, 14, 15].

Region-based tracking is computationally efficient and works well in free-flowing traffic. However, under congested traffic conditions, vehicles partially occlude one another instead of being spatially isolated, which makes the task of segmenting individual vehicles difficult. Such vehicles will become grouped together as one large blob in the foreground image. These methods cannot usually cope with complex deformation or a cluttered background.

2.3 Contour-based tracking

Contour-based tracking is dual to the region-based approach. 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.

These algorithms provide more efficient description of objects than do region-based algorithms, and the computational complexity is reduced. However, the inability to segment vehicles that are partially occluded remains. If a separate contour could be initialized for each vehicle, the tracking could be done even in the presence of partial occlusion. For all methods, initialization is one of the major problems.

2.4 Feature-based tracking

Feature-based tracking abandons the idea of tracking objects as a whole, but instead tracks features such as distinguishable points or lines on the object. Even in the presence of partial occlusion, some of the features of the moving object remain visible, so it may overcome the problem. Furthermore, the same algorithm can be used for tracking in daylight, twilight or night-time conditions, as well as different traffic conditions. It is self-regulating because it selects the most salient features under the given conditions (e.g. window corners, bumper edges... during the day and tail lights at night).

Tracking features is done through well developed methods such as Kalman filtering and the Kanade-Lucas-Tomasi Feature Tracker [2]. Since a vehicle can have multiple features, the next problem is the grouping or clustering of features, i.e. deciding what set of features belongs to the same object. The main cues for grouping are spatial proximity and common motion. These algorithms can adapt successfully and rapidly, allowing real-time processing and tracking of multiple objects in dense traffic.

3 Feature-based tracking in intersections

As far as we know, the main approaches for intersections are not feature-based, but mainly region-based [7, 10, 11, 15]. However, feature-based tracking is preferred since it can handle partial occlusions, which is a major problem in intersections [7]. Feature tracking is addressed by the robust Kanade-Lucas-Tomasi feature tracker [2]. The system input is a set of feature tracks, i.e. temporal series of coordinates, and the problem is to group them to generate vehicle hypotheses. Using the assumption that road surfaces are flat and that vehicle motions are parallel to the road plane, the world coordinates of the features can be retrieved by computing an homography between the image coordinates and the world coordinates.

3.1 Grouping features

Some approaches group the features independently in each frame [3, 8]. A simple grouping method grows nearest neighbor groups based on the distance and motion of features [3]. This is improved in [8] by using more sophisticated motion segmentation algorithms such as Normalized Cuts [13]. These solutions however lack temporal consistency, which makes the extracted trajectories noisy (See Figure 2).

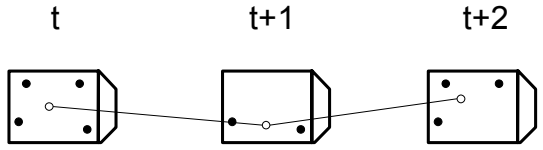

Figure 2: Illustration of the noise problem with methods that group features in each frame independently. The resulting trajectory of the centroid (the white dot) of the feature group (the black dots) shows a wrong vehicle movement.

Other approaches [1] improve the resulting trajectories by using the whole feature tracks provided by their reliable feature tracking method. The constraint of common motion over trajectory lifetimes is used as the central cue for grouping. Features that are seen rigidly moving together are grouped together. The vehicle segmentation is achieved by integrating the spatial information over as many image frames as possible. To fool the algorithm, the two vehicles would have to have identical motions during the entire time they were being tracked.

Processing the whole feature tracks instead of the features in each frames can be seen as a global clustering of the feature tracks. Clustering a batch of feature tracks collected over a given period of time would be a difficult problem. This is tackled incrementally by grouping all nearby features together and then progressively segmenting the group over time based on the common motion constraint, which allows to use the method in real-time applications as demonstrated in [1].

However, [1] deals only with highways, where tracking is made easier by straight trajectories with occasional lane changing, a given entrance region and exit region, and complete tracks from the entrance region to the exit region. In intersections, the trajectories are more variable. Vehicles may stop before crossing the intersection, or even on the approaches. There are more than one detection and exit regions (there are for example four entrance and exit regions in the intersection displayed in Figure 1). In turning maneuvers, the vehicle pose will change, and features are often lost (see Figure 3). Feature tracks are also disrupted by occlusion caused by other vehicles and obstacles in the field of view such as poles (see Figure 4), trees...

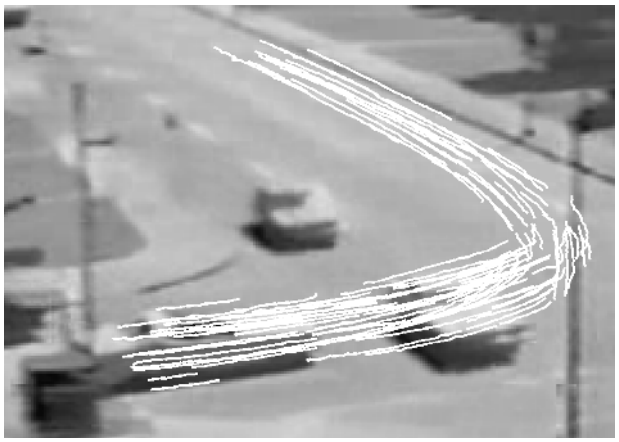

Figure 3: Illustration of feature tracking disruption as a vehicle is turning and its pose changes (displayed upon the frame of the first feature detection). However, the features are correctly grouped as one vehicle with our method.

In intersections, the feature tracks will often be "partial" due to the disruption problems. Features can be detected and lost anywhere in the field of view. Systems for vehicle tracking in intersections must address this problem. This is done by allowing the connection of features wherever they are detected.

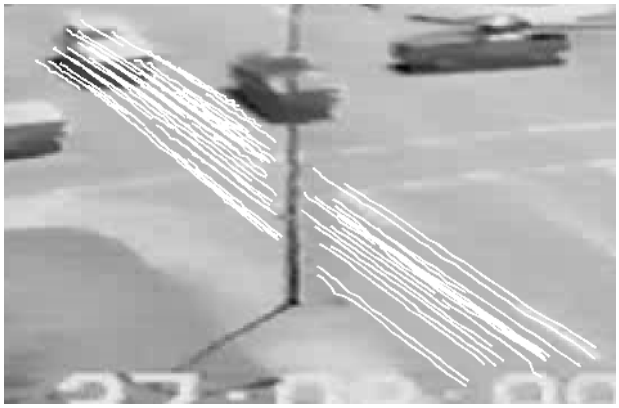

Figure 4: Illustration of feature tracking disruption caused by a pole (displayed upon the frame of the first feature detection). However, the features are correctly grouped as one vehicle with our method.

3.2 The algorithm

Some parameters have to be specified off-line. A special step involves the computation of the homography matrix, based on the geometric properties of the intersection or information provided with the video sequences. The algorithm relies on world coordinates, in which all the distances are computed. On the contrary to [1], the entrance and exit regions are not specified since feature grouping in intersections is not based on these information.

Our algorithm is adapted from [1]. The feature grouping is achieved by constructing a graph over time. The vertices are feature tracks, edges are grouping relationships between tracks and connected components (feature groups) correspond to vehicle hypotheses. For each image frame at time t,

1. Features detected at frame t are selected if they are tracked for a minimum number of frames, and if their overall displacement is large enough. Each newly selected feature f_i is connected to the currently tracked features within a maximum distance threshold $D_{connection}$.

2. For all pairs of connected features (f_i, f_j) that are currently tracked, their distance $d_{i,j}(t)$ is computed and their minimum and maximum distances are updated. The features are disconnected (i.e. the edge is broken in the graph) if there is enough relative motion between the features, formally if

$$max_t\ d_{i,j}(t) - min_t\ d_{i,j}(t) > D_{segmentation} \quad (1)$$

where $D_{segmentation}$ is the feature segmentation threshold.

3. The connected components in the graph are identified. Each connected component, i.e. set of feature tracks, is a vehicle hypothesis. If all the features that compose a component are not tracked anymore, the features are removed from the graph and the characteristics of the vehicle hypothesis are computed (centroid position, speed vector, vehicle size).

Only the minimum and maximum distances between the connected features are stored and updated at each time step, which reduces the computational complexity. The main parameters are the two thresholds $D_{connection}$ and $D_{segmentation}$. When features are detected, they will be connected to many other features. Vehicles are overgrouped then. As vehicles move, features are segmented as their relative motion differs. When a connected component of the graph is only composed of features that are not tracked anymore, a vehicle hypothesis is generated.

The main difference with respect to [1] is that there is no assumption with respect to entrance or exit regions so that partial feature tracks can be used. When features are detected or lost, this is taken into account, even though they may not be located at this time near the border of the field of view.

For that purpose, it must be ensured that the common motion constraints between all connected features can be correctly assessed. Once a feature is not tracked anymore, there is no way to disconnect it from the features that are still tracked. To avoid connecting features that don't move together but happen to be close when one is detected and the other is about to be lost, the features need to be both tracked over a minimum number of common frames.

A balance must be found between oversegmentation and overgrouping. In our case, the choice of $D_{connection}$ is more important since one should not connect faraway features that move together but don't belong to the same vehicle. Only one feature is enough to connect to separate vehicle hypotheses and this should be avoided.

4 Experimental Results

The performance of the system is assessed on a variety of video sequences, recorded on four different intersections. The main sequences come from an old set of examples used to train traffic conflict observers [12] ("Conflicts" set). Two other sequences recorded on two different locations are taken from the repository of the Institut fr Algorithmen und

Kognitive Systeme of the University of Karlsruhe[1] ("Karlsruhe" set) and the last sequence can be found on Vera Kettnaker's former research webpage[2], used in [4] ("Cambridge" sequence). The lengths of the sequences are given in Table 1. The ground truth is not provided with these video sequences and the results are assessed manually.

Sequences	Length (frames)
Conflicts	5793
Karlsruhe	1050
Cambridge	1517

Table 1: Video sequences for evaluation, with their length (number of frames).

No preprocessing was done to suppress shadows or to stabilize occasional camera jitter. The parameter values were easily found by trial and error. As world coordinates are used, the parameters are the same for all sequences. $D_{connection}$ and $D_{segmentation}$ were set respectively to 5 meters and 0.3 meters. The Cambridge sequence is an exception because there is no homography information with the sequence, and the geometry of the intersection cannot be guessed from the view. However, since the vehicle scale is preserved in most of the field of view, a simple factor was applied to the image coordinates so that the vehicle sizes were similar to the other sequences, and the same parameters values could be employed.

The results are displayed in the table 2. A true match is a matching between a vehicle and a group. A false negative is an unmatched vehicle. An overgrouping is counted if a group matches more than one vehicle. A false positive is a unmatched group. An oversegmentation is counted if a vehicle matches more than one group. The overall results are satisfying, with an average percentage of correctly detected vehicles of 88.4%. Consecutive tracking results are shown in Figure 5. Other successful feature groupings under difficult conditions were displayed in the Figures 3 and 4. Additional materials can be accessed online[3].

Many errors occur in the far distance, where small feature tracking inaccuracies imply larger error on the world coordinates. Other errors are caused by camera jitter. Most pedestrians and two-wheels are correctly tracked. There are more problems for larger vehicles such as trucks and buses, that are often oversegmented. Overgrouping happens when two vehicles move together or one feature is detected as moving consistently with two other distinct groups. More knowledge can be added to solve these problems, such as models of the vehicles, to begin with a threshold on the vehicle sizes. However, this type of model-based approach is rarely generic. Extra cues can be based on background subtraction or direct vehicle recognition.

Overgrouping or oversegmentation should be limited, but vehicles are still correctly detected when these errors occur. These are less serious problems than false positives or negatives, in which cases vehicles are either falsely detected or missed. Many applications can deal with a little oversegmentation or overgrouping.

[1] http://i21www.ira.uka.de/image_sequences/

[2] http://www.cs.rpi.edu/~kettnv/Research.htm

[3] http://www.confins.net/saunier/

5 Conclusion

This paper has presented an extension to intersections of the well-known feature-tracking approach described in [1]. Our method can handle partial feature tracks and more complex scenes, with multiple entrance and exit regions. Its performance is assessed on a variety of video sequences recorded on four different intersections. The algorithm will be improved for better error handling. More cues can be added for that purpose, for example based on background subtraction and direct vehicle recognition.

Acknowledgements: We thank Stan Birchfield for his implementation of the Kanade-Lucas-Tomasi feature tracker, and his student Neeraj Kanhere for sharing their code and their valuable comments.

References

[1] D. Beymer, P. McLauchlan, B. Coifman, and J. Malik. A real-time computer vision system for measuring traffic parameters. In *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, pages 495–501, Washington, DC, USA, 1997. IEEE Computer Society.

[2] S. T. Birchfield. Klt: An implementation of the kanade-lucas-tomasi feature tracker. `http://www.ces.clemson.edu/~stb/klt/`.

[3] S. T. Birchfield. *Depth and Motion Discontinuities*. PhD thesis, Stanford University, June 1999.

[4] M. Brand and V. Kettnaker. Discovery and segmentation of activities in video. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 22(8):844–851, August 2000.

Sequences	True Match	Overgroup.	False Neg.	Overseg.	False Pos.
Conflicts	215	27	8	33	6
	86.0%	10.8%	3.2%	13.0%	2.4%
Karlsruhe	36	1	1	7	0
	84.7%	2.6%	2.6%	16.3%	0.0%
Cambridge	51	2	1	7	2
	94.4%	3.7%	1.9%	11.7%	3.3%

Table 2: Tracking results for each sequence set. The rates every two row are computed by dividing the number of true matches, overgroupings and false negatives by their sum, and by the number of oversegmentations and false positives by their sum plus the number of true matches.

[5] A. Cavallaro, O. Steiger, and T. Ebrahimi. Tracking video objects in cluttered background. *Circuits and Systems for Video Technology, IEEE Transactions on*, 15(4):575– 584, April 2005.

[6] H. Dahlkamp, A. Ottlik, and H.-H. Nagel. Comparison of edge-driven algorithms for model-based motion estimation. In *Proceedings of the Workshop on Spatial Coherence in Visual Motion Analysis (SCVMA-2004)*, 2004.

[7] S. Kamijo, Y. Matsushita, K. Ikeuchi, and M. Sakauchi. Traffic monitoring and accident detection at intersections. *IEEE Transactions on Intelligent Transportation Systems*, 1(2):108–118, June 2000.

[8] N. K. Kanhere, S. J. Pundlik, and S. T. Birchfield. Vehicle segmentation and tracking from a low-angle off-axis camera. In *IEEE Conference on Computer Vision and Pattern Recognition*, San Diego, California, June 2005.

[9] D. Magee. Tracking multiple vehicles using foreground, background and motion models. *Image and Vision Computing*, 22:143–155, 2004.

[10] B. Maurin, O. Masoud, and N. P. Papanikolopoulos. Tracking all traffic: computer vision algorithms for monitoring vehicles, individuals, and crowds. *Robotics & Automation Magazine, IEEE*, 12(1):29–36, March 2005.

[11] S. Messelodi, C. M. Modena, and M. Zanin. A computer vision system for the detection and classification of vehicles at urban road intersections. *Pattern Analysis & Applications*, 8, 2005.

[12] N. Saunier and T. Sayed. Automated Road Safety Analysis Using Video Sensors. Technical report, University of British Columbia, 2006.

[13] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 22(8):888–905, August 2000.

[14] C. Stauffer and E. Grimson. Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 22(8):747–757, August 2000.

[15] H. Veeraraghavan, O. Masoud, and N.P. Papanikolopoulos. Computer vision algorithms for intersection monitoring. *IEEE Transactions on Intelligent Transportation Systems*, 4(2):78–89, June 2003.

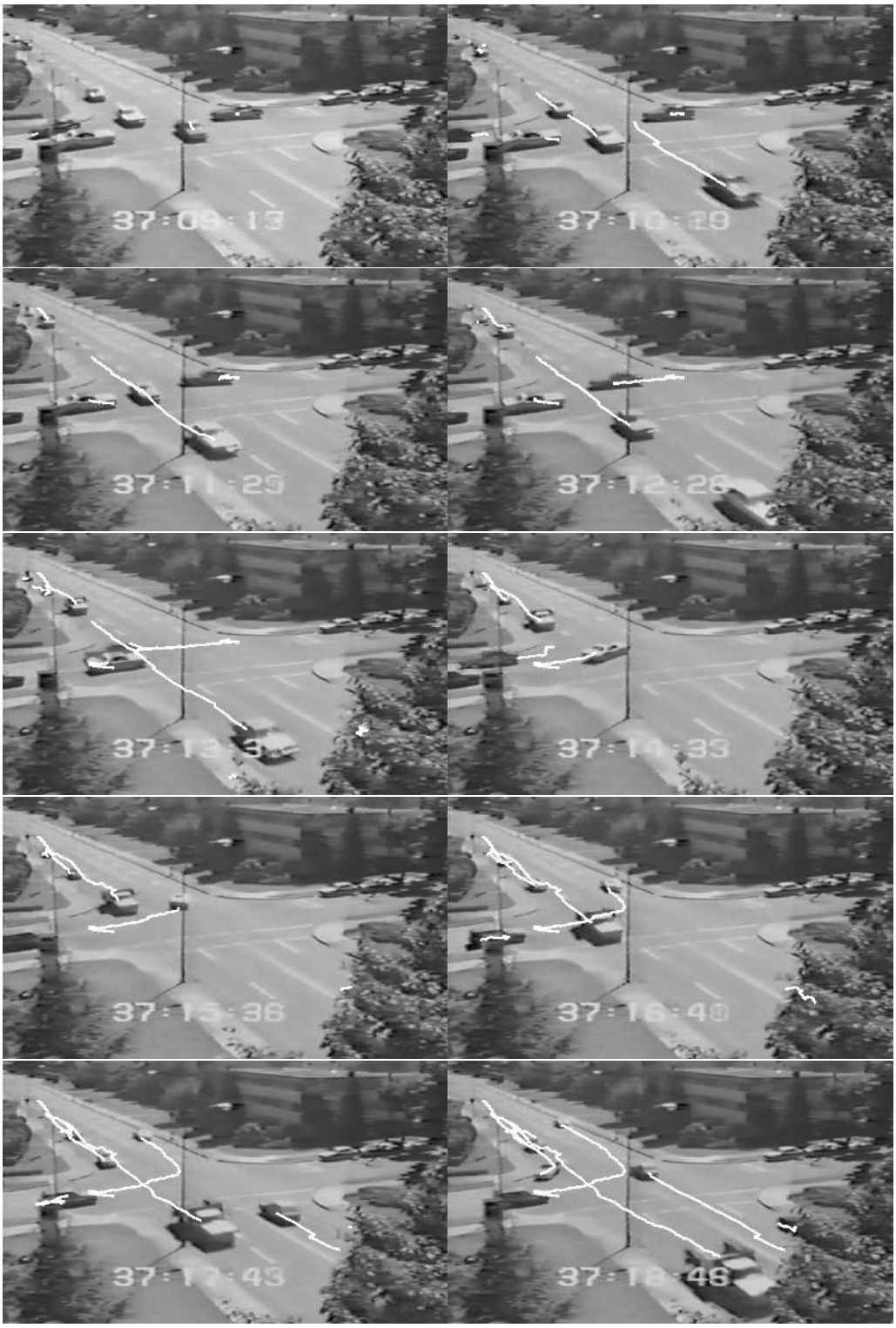

Figure 5: Sequence of image frames with the vehicle tracks overlayed, every 31 frames, from left to right and top to bottom.