# 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 -- http://www.confins.net/saunier/

## 1. Motivation

Need for **automated traffic monitoring** systems to assist human operators (traffic congestion and safety). Detailed traffic monitoring involves **vehicle tracking**. **Video sensors** have distinct advantages:
- they are easy to use and install,
- they offer rich traffic description (vehicle tracking),
- they can cover large areas,
- they are cheap sensors.

**Highways** have attracted considerable attention, at the expense of other parts of the road network, as **intersections**. Maurin *et al.* [4] state that despite significant advances in traffic sensors and algorithms, modern monitoring systems cannot effectively handle busy intersections.
Specific problems for vehicle tracking in intersections include 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. Other problems are common to highways and intersections: global illumination variations, multiple object tracking and shadow handling.

## 2. Related Work

There are 4 main approaches for object tracking:
- 3D **model**-based tracking,
- **region**-based tracking,
- **contour**-based tracking,
- **feature**-based tracking.

Features such as distinguishable points or lines on the object are tracked. This is achieved through well known methods such as the Kanade-Lucas-Tomasi Feature Tracker (KLT).
The problem is the **grouping** or **clustering** of the features.
The feature-based tracking approach is:
- robust to partial occlusion,
- self-regulating as it selects the most salient features under variable lighting conditions.

Feature-based algorithms can adapt successfully and rapidly, allowing real-time processing and tracking of multiple objects in dense traffic.

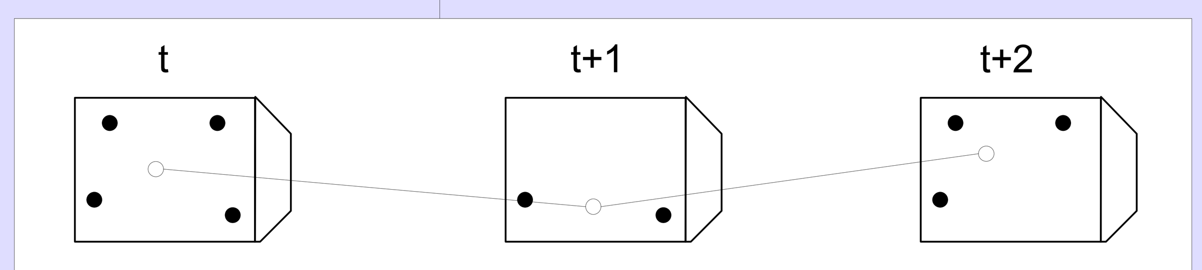
## 3. Feature Grouping

Feature grouping is deciding what set of features belong to the same vehicle. The main grouping cues are:
- **spatial proximity**,
- **common motion**.

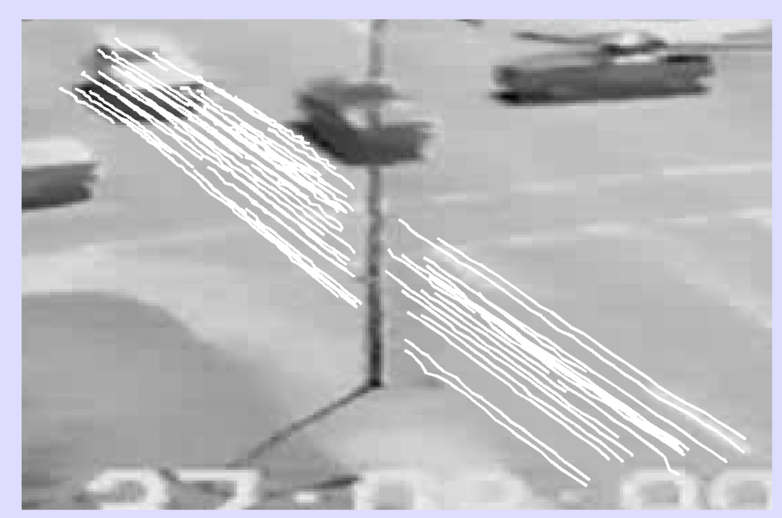
Grouping can be done independently in each frame (with nearest neighbours group growing [2] or more sophisticated motion segmentation algorithms such as Normalized Cuts [3]). This approach lacks temporal consistency.

Set of feature tracks $(x_1, y_1, \dots, x_n, y_n)$ (world coordinates)

Vehicles hypotheses




The alternative is to use whole feature tracks, generated by the feature tracking algorithms. Beymer *et al.* [1] use the constraint of **common motion over trajectory lifetimes**, but only deal with highways. This work presents an extension to intersections where
- trajectories are variable, with turns and stops,
- there are more than one entrance and exit regions for vehicles,
- features tracks are frequently disrupted (change of vehicle pose while turning, occlusion).



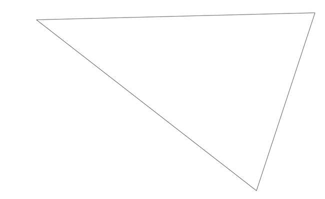
## 4. Feature Grouping Algorithm

construction of a **non-oriented graph** over time

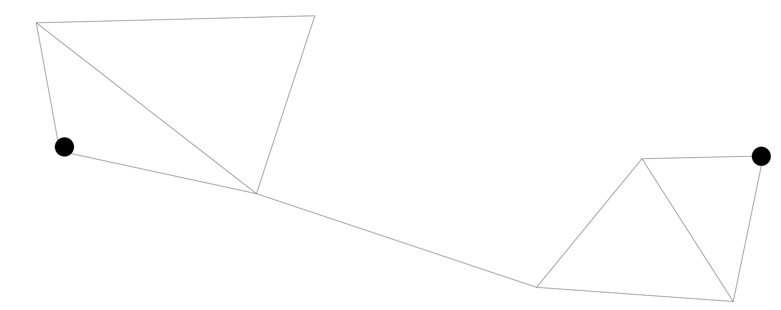
vertex: feature track

edge: grouping relationship

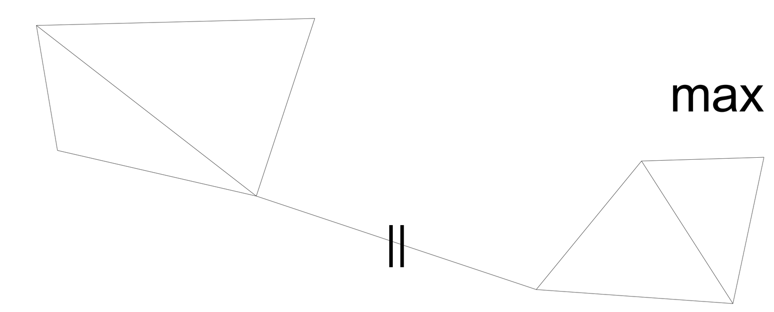
$d_{ij}(t)$: distance between feature i and j at time t



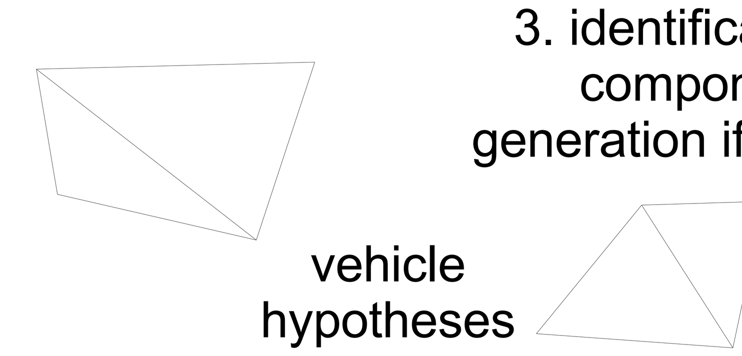
For each image at time t,



1. connection of recently detected features within distance $D_{connection}$




2. feature disconnection if relative motion is too large $\max d_{ij}(t) - \min d_{ij}(t) > D_{segmentation}$



3. identification of the graph connected components and vehicle hypothesis generation if the features are not tracked

vehicle hypotheses

When features are detected, they will be connected to many other features. Features are **overgrouped** then. As vehicles move, their relative motion differs and features are **segmented**. To ensure that connected features can be disconnected, the two features must be tracked simultaneously for a minimum number of frames. The main difference with the work presented in [1] is that there is no assumption made on entrance and exit regions so that partial feature tracks can be used.

## 5. Experimental Results

Errors occur in the far distance, because of camera jitter, and for larger vehicles (bus, trucks) that are often oversegmented.

| Sequences | Length (frames) | True Match | Overgroup. | False Neg. | Overseg. | False Pos. |
|---|---|---|---|---|---|---|
| Conflicts | 5793 | 215 | 27 | 8 | 33 | 6 |
| | | 86.0% | 10.8% | 3.2% | 13.0% | 2.4% |
| Karlsruhe | 1050 | 36 | 1 | 1 | 7 | 0 |
| | | 84.7% | 2.6% | 2.6% | 16.3% | 0.0% |
| Cambridge | 1517 | 51 | 2 | 1 | 7 | 2 |
| | | 94.4% | 3.7% | 1.9% | 11.7% | 3.3% |


## Conclusion

This method can handle partial feature tracks and complex traffic scenes with multiple entrance and exit regions, such as intersections. The algorithm is currently re-implemented for better performance with Intel OpenCV library. More cues can be added for better results, for example based on background subtraction and direct vehicle recognition.

## References

1. D. Beymer, P. McLauchlan, B. Coifman, and J. Malik. A real-time computer vision system for measuring traffic parameters. In IEEE Conference on Computer Vision and Pattern Recognition, 1997.
2. S. T. Birchfield. Depth and Motion Discontinuities. PhD thesis, Stanford University, 1999.
3. 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, 2005.
4. B. Maurin, O. Masoud, and N. P. Papanikolopoulos. Tracking all traffic: computer vision algorithms for monitoring vehicles, individuals, and crowds. Robotics & Automation Magazine, 2005.