Rule-based Reasoning on Visual Data for Urban Traffic Monitoring

R. Cucchiara[1], M. Gavanelli[2], A. Prati[1], M. Piccardi[2]
1 DSI, University of Modena, Via Campi 213/b - 41100 Modena, Italy
2 DI, University of Ferrara, Via Saragat 1 - 44100 Ferrara, Italy

Abstract

The paper describes a system for detecting vehicles in urban traffic scenes by means of rule-based reasoning on visual data. The strength of the proposed approach is its formal separation between low-level image processing modules (able for extracting visual data under various illumination conditions) and the high-level module, which provides a single framework for tracking vehicles in the scene. The image processing modules extract visual data from the scene, by spatio-temporal analysis during day-time, and by morphological analysis of headlights at night. The high-level module is designed as a forward chaining production rule system, working on symbolic data, i.e. vehicles and their attributes (area, pattern, direction...) and exploiting a set of heuristic rules tuned to urban traffic conditions. The synergy between the artificial intelligence techniques of the high level and the low-level image analysis techniques provides the system with flexibility and robustness.

1 Introduction

Intelligent Transport Systems (ITS) is a wide research area where Artificial Intelligence techniques are applied on traffic data [Masaki, 1998]. Traffic data may come from different sensors such as inductive loops, pneumatic systems, or cameras. In particular, in the area of urban traffic monitoring, real-time data are integrated with short and long-term knowledge on the traffic status, in order to dynamically update traffic information.

It is undoubtable that vision-based camera systems are more sophisticated and powerful than those based on other sensors, since the information content associated with image sequences allow many inferences on vehicle tracking and classification. However, many existing vision-based systems are not capable of providing detailed information on individual vehicles but are limited to measure or quantify the traffic flow only, or to solve specific sub-problems (e.g. queue detection [Aubert *et al.*, 1996], inductive loop emulation [Michalopoulos, 1991], congestion detection on highways [Beymer and Malik, 1996]), lacking generality. Traffic control should be adaptable to different environment, weather and light condition; in addition, urban scenes are particularly complex since the background condition is highly variable [Koller *et al.*, 1994].

The approach we propose defines a general-purpose framework for traffic monitoring. Our approach is based on a two-level system, where a high level production rule-based reasoning system supervises different low-level image processing modules. The interface between the two levels is the symbolic datum *Vehicle*. Fig. 1 sketches the overall VTTS

(Vehicular Traffic Tracking System) architecture, showing the main processing steps in the high and low-level modules.

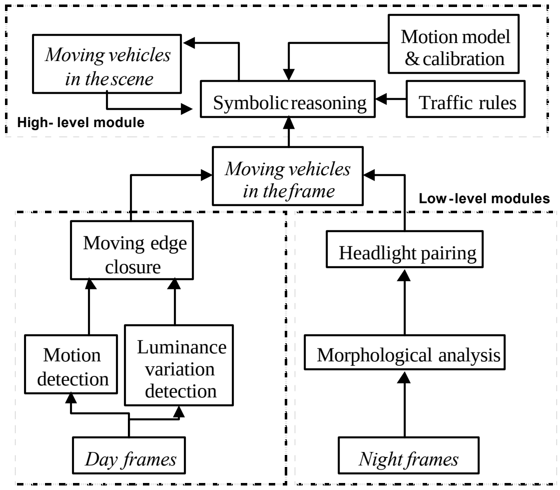

Fig. 1 The VTTS system architecture

The high-level system exploits forward-chaining reasoning for producing inference on moving vehicles: it's goal is to perform vehicle tracking and correct potential errors of the low-level module. It copes with over-segmentation (a vehicle split into more parts), under-segmentation (more vehicles, partially overlapped, merged into only one), false detection, and miss-segmentation, due to luminance condition variations, shadows and non-ideality of scene. The low-level modules extract vehicles from day-time and night image sequences, suitably matching the different perceivable objects: during the day, moving templates are detected as vehicles; instead, at night, target objects are the vehicle headlights. These different image processing tasks have the common goal of extracting moving vehicles and their attributes.

2 The high-level tracking module

The high-level module has been conceived for managing symbolic visual data extracted from the scene, performing various tasks related to traffic management. In the current prototype, these tasks include the number of moving and road-crossing vehicles, and the queue length during the red-light phase; other information such as traffic congestion can be inferred, based on heuristic rules. All of this information is of interest for a distributed traffic light controller, as the one working in Bologna [Utopia, 1997].

In this work, we adopt a general purpose reasoning system for tracking: a production rule system with forward chaining [Hyes Roth, 1983], formalizing the environment knowledge and the relationships between data extracted by the image processing modules. The symbolic reasoning system is based on a working memory and a production rule set.

The basic symbol of the working memory is the entity Vehicle, described in terms of a list of attributes:

Vehicle =(Id, Extent, Est_Ext, Dir, Displ, Tr_Frames, Lost_n, Stop_n, Status).

Id is the vehicle's identifier, *Extent* and *Est_Ext* are the shape extents computed at the current frame and estimated for the next one, based on the displacement (*Disp*) between the last two vehicle's positions, the direction (*Dir*) and the perspective variation. *Tr_Frames* states since how many frames the vehicle is being tracked.

The goal of the system is to track both moving and stationary vehicles, i.e. vehicles in motion in previous frames, but not detected as moving objects by the low-level modules at current frames. Instead, vehicles leaving the scene must not be kept the working memory anymore. To this aim, other attributes are added: *Status* (that can be moving or stopped), *Lost_n* stating since how many frames the vehicle has been lost by the low-level module, and *Stop_n* since how many frames the object has been classified as stopped.

The working memory is composed of two parts: the first is given by the set VS=[$Vs^{k-1}_1 \ldots Vs^{k-1}_n$] of all n *vehicles* in the scene (computed up to the k-1 frame), the second given by the set VM=[$Vm^k_1 \ldots Vm^k_m$] of m *moving vehicles*, obtained in real-time by the low-level module with respect to the current k frame.

The defined production rules aim to verify the vehicle presence between frames, by considering elements in sets *VS* and *VM* in order to create a new set *NEXT*_VS=[$Vs^k_1 \ldots Vs^k_p$] that will upgrade the knowledge at the k^th frame. The p number of elements in *NEXT_VS* satisfies $p \leq n+m$. At the end of the rule-based reasoning, the assignment $VS \leftarrow NEXT\_VS$ will be performed.

The basic *match(X,Y)* rule considers several criteria for assessing that *X* and *Y* are the same vehicle, being *X* in *VS* set, and *Y* in *VM* set: the *closeness* between the estimated new position of *X* and the computed position of *Y*, the *area-similarity* between extents (corrected by the perspective), and the *pattern-similarity*. If the rule $match(Vs^{k-1}_i, Vm^k_j)$ is verified, it updates the attributes of old vehicle Vs^{k-1}_i , which is then inserted in the new *NEXT_VS* set.

match(Vsk-1i,Vmkj) $\Leftarrow$ *(Is_In_VS(Vsk-1i), Is_In_VM(Vmkj), close_position(Vsk-1i,Vmkj),*

equal_area(Vsk-1i,Vmkj), equal_pattern(Vsk-1i,Vmkj)[1]

NEXT_VS(Vsk-1i) $\Leftarrow$ *match(Vsk-1i,Vmkj)*

The computation of the last fact - *equal_pattern* - in the rule is highly time-consuming, since it involves pixel-level correlation. In particular correlation between extents is used, after adequate extent scaling to normalize their areas. Due to its complexity, the *equal_pattern(X,Y)* is not always checked, but only in the case of vehicles lacking motion information.

---

[1] We use a Prolog-like notation, where the comma has the meaning of AND connective

The previous rule is adequate for tracking only if we do not consider "born" and "dead" vehicles, which are respectively those entering the scene or disappearing. This last situation is due either to objects leaving the scene (for instance, at a green light) or objects stopping at a red light (stopped objects are not segmented by the motion detection algorithms). Nevertheless, in this last case vehicles should be tracked and some specific rules are added to provide heuristics for initial and final conditions.

The high-level module also has the goal of correcting low-level segmentation errors. Some of the rules allow for correcting over/under segmentation errors by deciding if two objects have to be merged into only one (in the case of over-segmentation) or to split one object into two (due to vehicles occlusion or under-segmentation). For instance, we include the following rules for storing vehicles in the *NEXT_VS* set:

$$NEXT\_VS(Vm^k_j, Vm^k_h)\ \ddot{U}(Is\_In\_VS(Vs^{k-1}_i), Is\_In\_VM(Vm^k_j), Is\_In\_VM(Vm^k_h), \\ overlap\_extent(Vs^{k-1}_i, Vm^k_j, Vm^k_h), not\_equal\_direction(Vm^k_j, Vm^k_h))$$

$$NEXT\_VS(Vs^{k-1}_i, Vs^{k-1}_l)\ \ddot{U}(Is\_In\_VS(Vs^{k-1}_i), Is\_In\_VS(Vs^{k-1}_l), Is\_In\_VM(Vm^k_h), \\ overlap\_extent(Vm^k_h, Vs^{k-1}_i, Vs^{k-1}_l), not\_equal\_direction(Vs^{k-1}_i, Vs^{k-1}_l))$$

$$NEXT\_VS(Vm^k_j)\ \ddot{U}(Is\_In\_VS(Vs^{k-1}_i), Is\_In\_VS(Vs^{k-1}_l), Is\_In\_VM(Vm^k_j), \\ overlap\_extent(Vm^k_h, Vs^{k-1}_i, Vs^{k-1}_l), equal\_direction(Vs^{k-1}_i, Vs^{k-1}_l))$$

$$NEXT\_VS(Vm^k_j, Vm^k_h)\ \ddot{U}(Is\_In\_VS(Vs^{k-1}_i), Is\_In\_VM(Vm^k_j), Is\_In\_VM(Vm^k_h), \\ overlap\_extent(Vs^{k-1}_i, Vm^k_j, Vm^k_h), equal\_direction(Vm^k_j, Vm^k_h))$$

In general, the information on opposite direction for two moving vehicles is stronger than other rules as shown by the first two rules. However, when the direction is the same and an overlap occurs, two old vehicles in VS are being detected as one, and in particular the element of VM which extent overlaps their extents. The third rule takes into account that two objects perceived separately in the past due to non-idealities (reflections, shadows..), could be two parts of the same vehicle. The fourth rule states that a single old vehicle is split in two new ones. This corrects the error of two objects being very close to each other in the past, due to over-segmentation, shadows or segmentation errors. Many other rules are included in the reasoning system for coping with miss-segmentation.

Experimental results show that these features allow for the correction of most segmentation errors, especially due to luminance variation during vehicle motion (e.g. when a vehicle drives through a building shadow) and provide robust traffic tracking.

3 The low-level image processing module

The role of the low-level image processing module is to extract visual data from the scene, in particular by forming hypotheses on vehicles to be submitted to the reasoning level.

Vehicle tracking must be performed during periods of interest for traffic monitoring and control, that usually range over very different illumination conditions. To this aim, we have defined two different sets of image analysis algorithms for extracting vehicles from image sequences either in day-time or at night. In daylight, information on motion and high gradient points are jointly exploited; instead, at night, features such as headlights are detected and associated with vehicles; headlights must be discriminated from reflections, beams, street-lamps and horizontal signals such as zebra crossings. Fig. 2 shows two frames acquired in daylight and at night, respectively, together with the extracted vehicles.

The algorithms for day-time vehicle extraction are [Cucchiara and Piccardi, 1999]:

- detection of moving points by performing a difference on three consecutive frames;
- detection of high contrast points in the image, i.e. points with high gradient, as possible edges of moving objects;
- execution of a Moving Edge Closure, that is a morphological closure between moving points and sharp edges in order to extract moving objects.

The moving points detection is based on a double-difference image operator performing thresholded difference between three frames, in order to segment only "strong" moving points of true moving objects [Yoshinari and Michihito, 1996]. The operator filters isolated spots due to small movements of sensors and avoids de-localization of extracted points, and furthermore is more powerful than two-frame difference or difference-with–background techniques used in other systems [Koller *et al.*, 1994]. The limit we have found to appreciate motion in urban traffic scenes is to sample one frame in every five and group these frames in sequences of three for performing the double-difference operation.

(a) Source frame in day-time | (b) Extracted vehicles

(c) Source frame at night	(d) Extracted vehicles

Fig. 2 Vehicle detection in day-time and at night

The zone around moving points is used as a local mask where the luminance gradient is computed. The gradient is exploited together with the information on motion: in our approach we define a suitable *Moving Edge Closure*, in order to obtain a close contour of a moving object. Motion is the main property, and luminance gradient is exploited as complementary information: this means that the gradient assumes “high” or “low” values with respect to its average level in a small region of interest, making the system adaptive to luminance variation in time and space. Finally, a moving object is classified and labeled as a *vehicle* if its size (in pixels) is in accordance with an initial scene calibration. This light-cost initial calibration includes masking of the inspected area, and drawing of a (poly)line on the lane that represents the main traffic direction. All metric parameters used for vehicle extraction, such as distances and sizes, are scaled linearly along the main traffic direction.

At night, the approach differs substantially: headlights must be separated from beams, but their motion vectors are similar. In addition, the sampling rate allowed by standard cameras (25/30 frames per second) does not allow to easily estimate motion of an object with respect to its previous positions, since objects are too small to overlap in two consecutive frames at common vehicle speeds. Therefore, the algorithms for vehicle extraction at night are not based on motion detection, but mainly *aim to identify headlight shapes in each frame*.

The algorithms for night vehicle extraction are [Cucchiara and Piccardi, 1999]:

- image thresholding (the bimodal histogram allows easy separation of objects from background);
- template matching on the image with a headlight template, scaled in proportion to the image region;
- cross-correlation on headlight pairs, with correlation parameters scaled in proportion to the image region.

The last step aims to make vehicle detection more robust while discriminating vehicles from motorbikes, since they have a different impact on typical traffic flow parameters such as throughput, congestion probability, and queue length, and different relevance for traffic management strategies. This verification is based on correlation between headlights belonging to a same pair; correlation is performed by matching luminance values along the normal to the main traffic direction (taking perspective into account). The vehicle extent is assumed as the minimal rectangle including the headlight pair, plus a virtual vehicle body.

4 Experimental results

The tracking system has been evaluated on different real traffic scene sequences. Fig. 3 and Fig. 4 at the end of this paper show results in day-time with a sequence of 85 frame triples containing some vehicles moving in both directions. The graph of Fig. 3 measures low-level module performance: the “true” histogram reports the number of moving vehicles correctly extracted by the low-level system as a function of the frame number in the sequence. The other values report the false positives and false negatives: the former are objects different from vehicles but classified as vehicles, while the latter represent real vehicles that are not segmented. In this experiment, there are 188 real vehicles, 11 false negatives (5.8%) and 25 false positive (13.2%). False positives are more frequent, corresponding to over-segmentation errors and detection of small moving patterns like reflections; false negatives are mainly due to under-segmentation. Fig. 7 shows the results achieved by the high-level tracking module, reporting the number of ground-truth vehicles (objects that were in motion for at least three frames, that must be tracked) and the effectively tracked vehicles. The system tracks 332 vehicles of 343, with an error rate of 3.1%. This low error rate proves that the high-level system is able to substantially correct errors of the low-level module, achieving good results. Moreover, as shown in Fig. 7, all vehicles are tracked at least once during the sequence, and only a few are “lost” for one or two frames. Finally, it should be noted that the system is able to track also stopped vehicles (thanks to the STOPPED attribute): for instance, in frames 64-70 of Fig. 3, no moving vehicles are detected, while they are tracked in the same frames in Fig. 4.

Figg. 5-6 report results over a typical sequence at night, made of 140 frames. The total number of real vehicles present in the sequence frames is 631, with 29 false positives (4.5%) and 32 false negatives (4.9%) at the low level; instead, the tracking module shows a nearly zeroed error, if we consider the unavoidable delay of a few frames with respect to the low-level measurements.

5 Related works

Many previous works have proposed algorithms and systems for traffic monitoring. As a general remark, we judge that there are still some key issues open, and that this is confirmed by the relatively small number of systems installed in real application.

In the literature, two kinds of systems for video-based traffic control are outlined. The former ones rely on Intelligent Vehicle Highway Systems (IVHS - see for instance [Koller *et al.*, 1994; Smith *et al.*, 1996; Taktak *et al.*, 1995]), the latter ones on Urban Traffic Control systems (UTC – see for instance [Lipton *et al.*, 1998; Wixson *et al.*, 1998; Michalopoulos, 1991]). Although these systems aim to resolve similar problems, the techniques and the algorithms to be used vary significantly, and the effectiveness of the former group systems in the more complicated urban environment has yet to be proven.

A recent work from Lipton and others [Lipton *et al*., 1998] proposes a real-time, low-cost system for urban vehicle detection and tracking. A three-stage approach is proposed: a) extraction of moving points by inter-frame difference and clustering into moving regions by a connected component algorithm; b) only for targets persisting in time, classification into two categories - vehicles and human - based on a simple measure of dispersion; c) vehicle tracking: in frame Rn, candidate motion regions are compared with a vehicle in frame Rn-1, so as to establish the best correspondence. The vehicle is also compared with the same image zone in frame Rn, in case the object is no longer in motion. The approach proposed faces many of the key issues required by real operation, such as real time and low cost, and assesses several substantial improvements with respect to many previously proposed systems. The reported precision in tracking is rather high, but it could be improved by the use of more sophisticated algorithms, like the double-difference we adopt in this paper.

Furthermore, in [Lipton *et al*., 1998] illumination different from daylight is not considered. Instead, in [Taktak *et al*., 1995], automatic vehicle detection is performed using the same algorithms in the whole day, so there is no need for thresholding between day and night. Two markers are used at the same time: a) bright light, and b) light and dark blobs. Assessed precision is very high (96.40% in daytime, 98.05% at night 96.45% in twilight). However, no tracking system is used to compensate these errors, as we propose instead in this work, and only highway traffic is monitored, which is more easily monitored than urban traffic, particularly the detection of traffic circulation lane limits. Also [Wixson *et al*., 1998] addresses the problem of assessing illumination, in order to estimate the best algorithms and parameters. To this aim, all non-background elements are used to update various statistical measures of lighting, contrast, and shadows. Since the switching between daytime and night (dawn, dusk) is characterized by oscillations, the system we propose in this paper exploits similar measures in order to switch between daytime and night algorithms. This solution is more effective than static illumination assessment based on a calendar and timetable, such as used in Autoscope [Michalopoulos, 1991], which is hard to tune and rigid.

6 Conclusions

In this paper we have presented an approach to vehicle tracking based on rule-based reasoning on visual data. The reasoning module is designed as a forward-chaining production rule system, working on symbolic data and exploiting a set of heuristics tuned to urban traffic conditions. It is able to perform reliable vehicle tracking and compensate the segmentation errors made by the image analysis operators. Since vehicle tracking is required during different periods of the day, two sets of image analysis algorithms have been defined: in day-time, spatio-temporal analysis on moving templates is performed, while at night morphological analysis of headlight pairs. The overall system (VTTS - Vehicular Traffic Tracking System) has been tested on real road traffic scenes in the cities of Modena, Bologna, and Ferrara (Italy); experimental results prove that the integration between the supervisor and the image analysis techniques provides the system with flexibility and robustness.

This work is part of a project held with the Bologna Provincia government for a city control center with vision based traffic monitoring.

References

[Aubert *et al.*, 1996] D. Aubert, S. Bouzar, F. Lenoir, J. M. Blosseville. Automatic vehicle queue measurement at intersection using image processing. *Proc. of IEE Road Traffic Monitoring and Control*, pp. 100-104, 1996.

[Beymer and Malik, 1996] D. Beymer, J. Malik. Tracking vehicles in congested traffic. *Proc. of the 1996 IEEE Intelligent Vehicles Symposium*, pp. 130-135, 1996.

[Cucchiara and Piccardi, 1999] R. Cucchiara, M. Piccardi. Vehicle detection under day and night illumination. *Proc. of ISCS-IIA99*, Genoa, Italy, pp. 789-794, 1999.

[Hyes Roth, 1983] F. Hyes Roth, D. A. Waterman, D. B. Lenat. *Building expert systems*. Addison-Wesley, 1983.

[Koller *et al.*, 1994] D. Koller, J. Weber, T. Huang, J. Malik, G. Ogasawara, B. Rao, S. Russell. Towards robust automatic traffic scene analysis in real-time. *Proc. of the 33rd Conf. on Decision and Control*, Lake Buena Vista, FL, pp. 3776-3781, 1994.

[Lipton *et al.*, 1998] A. Lipton, H. Fujiyoshi, R. Patil. Moving target classification and tracking from real-time video. *Proc. of Fourth IEEE Workshop on Applications of Computer Vision WACV '98*, pp. 8-14, 1998.

[Masaki, 1998] I. Masaki. Machine-vision systems for intelligente transportation systems. *IEEE Intelligent Systems*, v. Nov/Dec 1998, pp. 24-31.

[Michalopoulos, 1991] P. G. Michalopoulos. Vehicle detection through video image processing: The AUTOSCOPE System. *IEEE Trans. on Vehicular Technology*, v. 40, n. 1, 1991, pp. 21-29.

[Smith *et al.*, 1996] E. Smith, C. A. Richards, S. A. Brand, N. P. Papanikolopoulos. Visual tracking for intelligent vehicle-highway systems. *IEEE Trans. on Vehicular Technology*, v. 45, n. 4, 1996, pp. 744-759.

[Taktak *et al.*, 1995] R. Taktak, R. Gresson, M. Dufaut, R. Husson. Vehicle detection for automatic traffic control during the whole day. *Proc. of First IFAC-Workshop on Advances in Automotive Control*, pp. 215-220, 1995.

[Utopia, 1997] *UTOPIA Urban Traffic Control, Technical Reference Manual*, Mizar Automazione, 1997.

[Wixson *et al.*, 1998] L. Wixson, K. Hanna, D. Mishra. Improved illumination assessment for vision-based traffic monitoring. *1998 IEEE Workshop on Visual Surveillance*, pp. 34-41, 1998.

[Yoshinari and Michihito, 1996] K. Yoshinari, M. Michihito. A human motion estimation method using 3-successive video frames. *Proc. of Int. Conf. on Virtual Systems and Multimedia*, pp. 135-140, 1996.

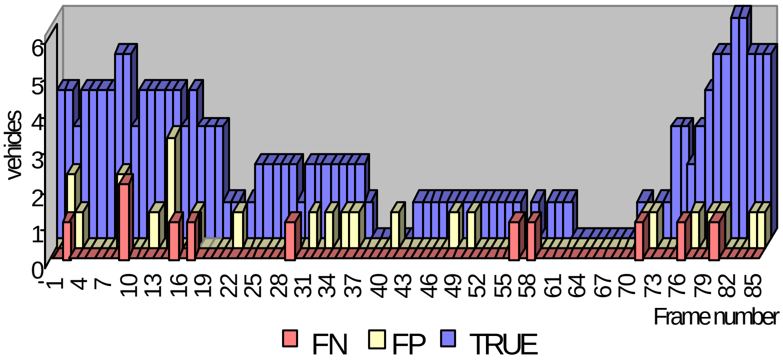

Fig. 3 Vehicles extracted by the low-level modules in the day-time

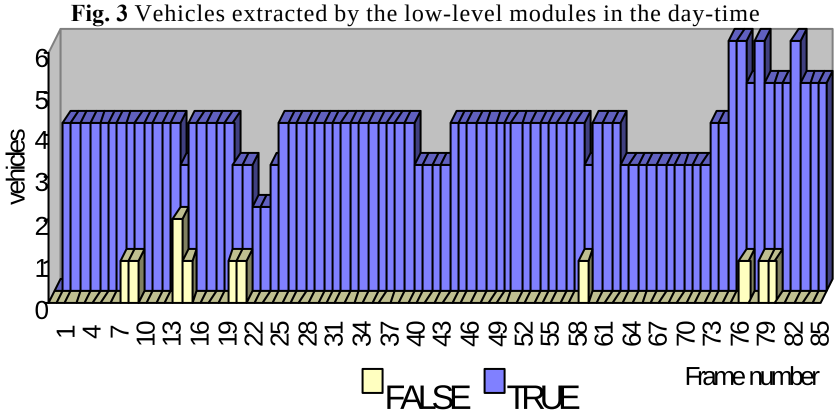

Fig. 4 Vehicles tracked by the high-level module in the day-time

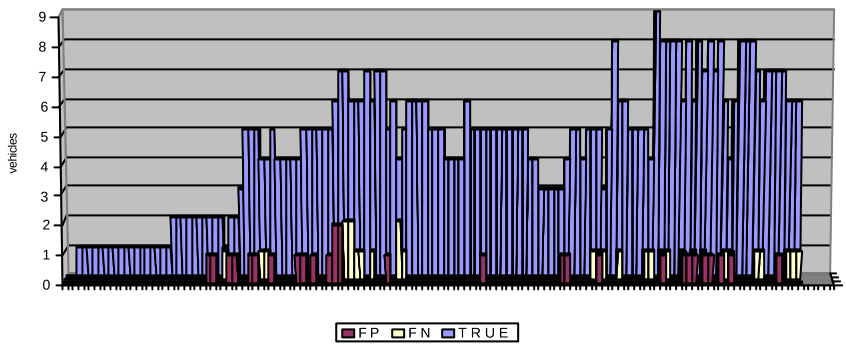

Fig. 5 Vehicles extracted by the low-level modules at night

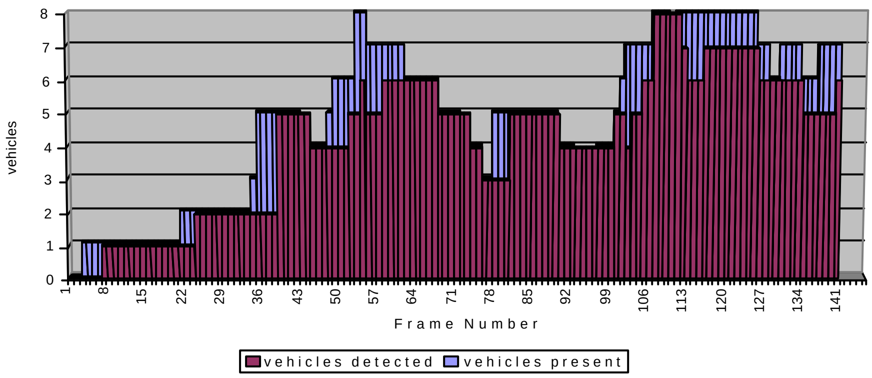

Fig. 6 Vehicles tracked by the high-level module at night