

Vehicle Detection under Day and Night Illumination

R. Cucchiara¹, M. Piccardi²

¹Dipartimento di Scienze dell'Ingegneria
Università di Modena e Reggio Emilia
Via Campi 213\b - 41100 Modena, Italy
e-mail: rita.cucchiara@unimo.it

²Dipartimento di Ingegneria - Università di Ferrara
Via Saragat 1 - 44100 Ferrara, Italy
e-mail: mpiccardi@ing.unife.it

Abstract

Effective detection of vehicles in urban traffic scenes can be achieved by exploiting image analysis techniques. Nevertheless, vehicle detection in daytime and at night can't be approached with the same image analysis algorithms, due to the strongly different illumination conditions. This paper describes the two different sets of image analysis algorithms that have been used in the VTTS system (Vehicular Traffic Tracking System) for extracting vehicles from image sequences acquired in daytime and at night. In the system, a supervising level selects the set of algorithms to apply and performs vehicle tracking under control of a rule-based decision module. The paper describes the tracking module, and reports experimental results for both vehicle detection and tracking.

I. INTRODUCTION

Automatic vehicle detection in traffic scenes is an important goal in the field of transportation systems, since it allows the enforcement of traffic policies with precise information on traffic. Image analysis provides several effective techniques to detect target objects in images, and has thus been extensively used for traffic monitoring systems.

However, many existing vision-based systems are not able 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 [1,2], inductive loop emulation [3], congestion detection on highways [3]...), lacking generality. Traffic control should be performed in different environment, weather and light conditions; in addition, urban scenes

are particularly complex since the background condition is highly variable [4].

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 based on the concept of *moving vehicle*. The low-level image processing modules compute moving vehicles, coping completely with different luminance conditions; instead, the high-level layer is independent of scene appearance, and reasons on symbolic data only.

The overall system (VTTS, Vehicular Traffic Tracking System) performs traffic tracking in urban roads and intersections under different luminance conditions.

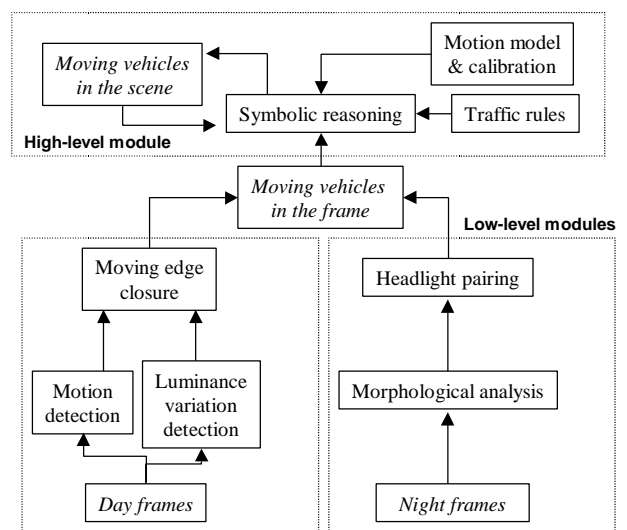


Fig. 1. The VTTS system architecture.

In this paper, we present different algorithms for day 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. The high-level system exploits forward chaining reasoning for producing inference on moving vehicles: its 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. Fig. 1 sketches the overall system architecture.

II. VEHICLE DETECTION IN DAY LIGHT

Reliable vehicle extraction is the basis for performing effective traffic monitoring; to this aim, the low-level image analysis modules perform vehicle extraction under the two main different illumination conditions: day light and night light.

Under day light, a traffic scene is made both of target vehicles and “distracting” objects, such as houses, trees, parked vehicles located along the lane (see Fig. 2).



Fig. 2 A traffic scene in day light.

In order to discriminate between these two categories of objects, a *mask* could be overlapped to the actual lane area, that is assumed to be the inspected zone. Nevertheless, this masking can not completely separate target objects from distractors; for instance, a same lane zone can be occupied by moving or parked cars at different times. Actually, motion seems to be the most discriminant feature of target objects and this is

confirmed by the experiments. In addition, the analysis of target objects’ spatial properties can enhance detection.

Therefore, target extraction is based on *spatio-temporal* segmentation: “temporal”, because it exploits information on moving points, extracted by relative difference between successive frames in the image sequence; “spatial”, because it takes into account luminance variations in the zone where motion is revealed [5].

The *moving points* detection (see Fig. 3) 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 [6, 7]. The operator filters isolated spots due to small movements of sensors and avoids de-localization of extracted points. It is more powerful than two-frame difference or difference-with-background techniques used in other systems [2,4].

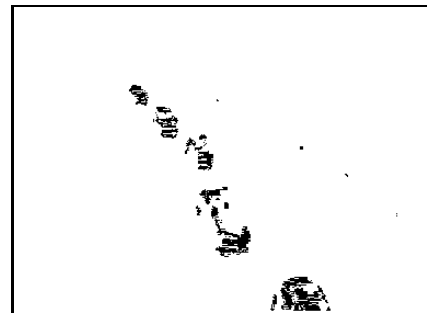


Fig. 3 Moving points in the scene.

The zone around moving points is used as a local mask where to compute the luminance gradient. 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 variations in time and space. A morphological closure of moving points is performed by following the points with high gradient; this defines the moving object contour; then, a chain code-based approach extracts the external edges only and is able to separate loosely connected objects with simple morphological operations (Fig. 4) [8].

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.

A further check is added for verifying possible overlays between objects: an analysis on the object circularity and other topological measures allows for dividing a single connected object in two or more ones.

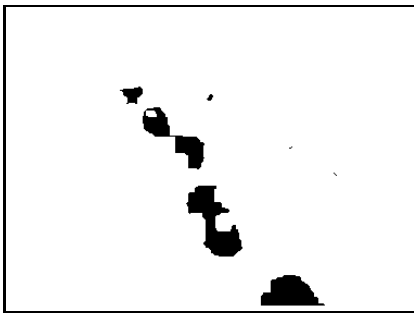


Fig. 4 Moving “objects”.

Moving objects are localized and characterized by their rectangular *extents*. Other proposals, for instance, make use of object’s external edges or snakes for representing vehicle [9]. This more complex segmentation is useful only if the object shape is important for tracking the trajectories and for an a-posteriori evaluation of possibly overlapped objects. We decided to adopt the extent only, in order to give the most compact representation to the high level systems.

III. VEHICLE DETECTION AT NIGHT

The scene appearance at night is very different from daytime; the only salient visual features are headlights and their beams, street-lamps, and horizontal signals such as zebra crossings (see Fig. 5). After masking the lane area, the scene becomes even simpler, since out-of-lane “distractors”, such as street-lamps, are removed.



Fig. 5 A traffic scene at night.

Our main goal is to identify vehicles in terms of *pairs of headlights*; this is a simplifying assumption, since vehicles with single headlight, such as motorbikes, can be present. Nevertheless, we neglect them, since they do not have significant impact on typical traffic flow parameters like throughput, congestion probability, and queue length.

Fig. 5 shows the inspected area (below the horizontal white line), where headlights’ pairs can be reliably detected; objects spanning the separation line are also removed.



Fig. 6 Non-linear luminance operator.

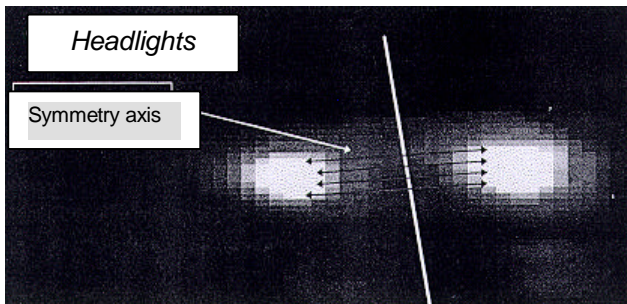
Since the image histogram is strongly bimodal, binarization can be easily achieved with non-linear operators such as thresholds or quadratic operators [8] (Fig. 6).

After these steps, we discriminate between headlights’ pairs and other objects, such as their beams, and horizontal signals. While all still objects could be separated by motion analysis, this is not true for beams. Thus, we decided to perform headlight detection via morphological analysis, by taking into account aspects like shape, size and minimal distance between vehicles; Fig. 7 shows the result of non-headlight removal.

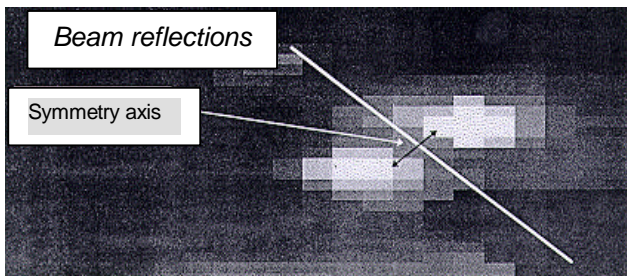


Fig. 7 Non-headlight removal.

Final 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. Fig. 8 shows a typical case of correlation: in Fig. 8a a pair of headlights is present, while in Fig. 8b beam reflections are shown. In the case of the real headlights, symmetry and luminance values are higher; the correlation operator is defined so as to exploit these properties, resulting much greater in the former case than in the latter; this allows for a further elimination of the disturbing beam reflections.



a)



b)

Fig. 8 Correlation on a) headlights and b) beam reflections.

Finally, the vehicle extent is assumed as the minimal rectangle including the headlight pair.

IV. THE TRACKING MODULE

Many works propose tracking algorithm for maintaining information on moving vehicles between frames. Some simply use trajectory filters, such as Kalman filters, in order to track trajectory [10].

In this work, we adopt a more complex reasoning system for tracking: a production rule system with forward chaining [11], formalizing both the traffic and 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 *vehicle*, described by a list of attributes

VEHICLE = (NAME, EXTENT, EST_EXT, DIR, DISPL, FRAME_N, LOST_N, STOP_N, STATUS).

NAME is the vehicle's identifier, EXTENT is the computed extent, while EST_EXT is the estimated future extent (in the next frame) computed combining the last vehicle movement (that is DISPL) and the prospective variation. DIR is the vehicle direction, since vehicles moving in both ways are tracked at the same time; FRAME_N indicates from how many frames the vehicle is tracked, LOST_N from how many it has been lost by the low level system, and STOP_N from how many the object is stopped. Finally STATUS is an attribute that can assume the values of MOVING or STOPPED. The extent is defined as:

EXTENT = (POSITION, AREA, PATTERN)

POSITION is the extent's 2D position in the image, AREA, is its number of pixels while PATTERN indicates the luminosity pattern, that is significant only for daytime moving vehicles.

The working memory is composed of two parts: the former is given by the old vector of *all vehicles* (V_x) in the scene, the latter containing the curr vector of *moving vehicles* (V_y), obtained in real-time by the low-level system.

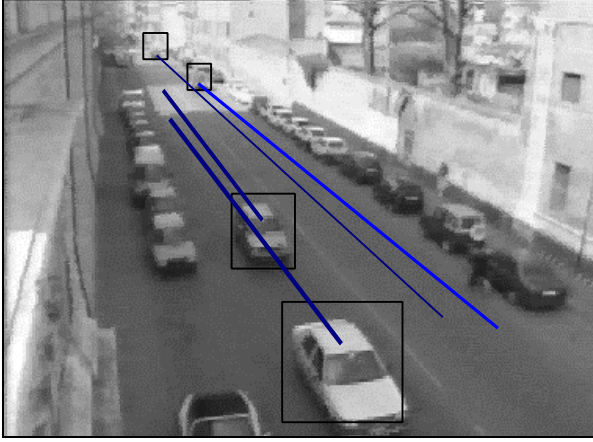


Fig. 9 Vehicle tracking.

The defined production rules aim at verifying the vehicle presence between frames. The basic rule considers several criteria: the *closeness* to the estimated new position, the *area-similarity* between extents (corrected by the perspective), and the *pattern-similarity*.

$$\text{match}(Vx, Vj) \leftarrow (\text{old}(Vx), \text{curr}(Vy), \text{close_position}(Vx, Vy), \text{equal_area}(Vx, Vy), \text{equal_pattern}(Vx, Vy))$$
¹

If a match between an old and a new vehicles is verified, the vehicle is kept in the old vector with updated attributes. Actually, the computation of the last fact in the rule is highly time-consuming, since it involves pixel-level correlation [8]. In particular correlation between extents is used, after adequate extent scaling in order to normalize their areas. Due to its complexity, the $\text{equal_pattern}(Vx, Vy)$ rule is not checked always, but only in the case of vehicles lacking motion information.

The previous rule is enough for tracking, only if we do not consider born and dead vehicles, which are the ones entering the scene or disappearing. This last situation is due either to objects leaving the scene (for instance, at a green light) or object switching light off during the night detection, or eventually objects stopping at a red light during the day (stopped objects are not segmented by the double-difference 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 system has also the goal of correcting low-level segmentation errors. Some of the rules allow for

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

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 ones (due to vehicles occlusion or under-segmentation).

For instance, we include the following rules for storing vehicles in the old vector in case of overlap:

$$\text{write}(Vy, Vk) \leftarrow (\text{old}(Vx), \text{curr}(Vy), \text{curr}(Vk), \text{overlap_extent}(Vx, Vy, Vk), \text{not equal_direction}(Vy, Vk))$$

$$\text{write}(Vx, Vy) \leftarrow (\text{old}(Vx), \text{old}(Vy), \text{curr}(Vk), \text{overlap_extent}(Vk, Vx, Vy), \text{not equal_direction}(Vx, Vy))$$

$$\text{write}(Vk) \leftarrow (\text{old}(Vx), \text{old}(Vy), \text{curr}(Vk), \text{overlap_extent}(Vk, Vx, Vy), \text{equal_direction}(Vx, Vy))$$

$$\text{write}(Vy, Vk) \leftarrow (\text{old}(Vx), \text{curr}(Vy), \text{curr}(Vk), \text{overlap_extent}(Vx, Vy, Vk), \text{equal_direction}(Vy, Vk))$$

In general, the information on opposite direction for two moving vehicles is stronger than other rules as shown by the first two rules. Instead when the direction is the same and an overlap occurs, a single old vehicle is split in two news, while the two old ones are merged in a single one. This means that two objects detected very close each other, could be due to over-segmentation, shadows or segmentation errors and these errors can be corrected by the current computation. Many other rules are included in the reasoning system for coping with miss-segmentation.

Experimental results show that these features allow for correcting most of the segmentation errors, especially due to luminance variation during vehicle motion (e.g. when it drives through a building shadow) and provide robust traffic tracking. Fig. 9 shows a traffic scene with superimposed tracking information.

V. EXPERIMENTAL RESULTS

The tracking system has been evaluated on different real traffic scene sequences. Fig. 10 and Fig. 11 show results with a sequence of 85 frame triples containing some vehicles moving in both directions. The graph of Fig. 10 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. 11 shows the results achieved by the high-level tracking system, 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. 11, 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. 10, no moving vehicles are detected, while they are tracked in the same frames in Fig. 11.

VI. CONCLUSIONS

In this paper we have presented two sets of image analysis algorithms to provide vehicle detection in traffic scenes. Two different approaches have been exploited, based on illumination conditions of day and night: in daytime images, spatio-temporal analysis on moving templates, while, in night images, morphological analysis of headlight pairs. A high-level module is used to supervise image analysis and provide vehicle tracking, designed as a forward-chaining production rule system, working on symbolic data and exploiting a set of heuristics tuned to urban traffic conditions. The overall VTTS system has been tested on image sequences from real traffic scenes: experimental results prove that the integration between the supervisor and the image analysis techniques provides the system with flexibility and robustness.

REFERENCES

- [1] Aubert, D., Bouzar, S., Lenoir, F., Blosseville, J.M. "Automatic Vehicle Queue Measurement at Intersection Using Image-Processing." IEE Road Traffic Monitoring and Control, Conference Publication no. 422, pp. 100-104 (1996).
- [2] Fathy, M., Siyal, M.Y. . "Real-time image processing approach to measure traffic queue parameters." IEE Proc.-Vis. Image Signal Process., vol. 142, no. 5, pp. 297-303 (1995).
- [3] Michalopoulos, P "Vehicle Detection video through image processing: the Autoscope system" IEEE Trans on cheichular technology 40-1 21-29, 1991.
- [4] Koller, D. Weber, J., Huang, T., Malik, J., Ogasawara, G., Rao, B., Russel, S. "Towards Robust Automatic Traffic Scene Analysis in Real-Time." Proc. Int'l Conf. Pattern Recognition, pp. 126-131 (1994).
- [5] Nakanishi, T, Ishii, K, "Automatic vehicle image extraction based on spatio-temporal image analysis", Proc. 11th ICPR, 1992, pp. 500-504.
- [6] Yoshinari K., Michihito M., "A human motion estimation method using 3-successive video frames" Proc. of Int. Conf. on virtual systems and multimedia, 135-140, 1996.
- [7] Barattin, M., Cucchiara, R., Piccardi, M., "A Rule-based Vehicular Traffic Tracking System", CVPRIP'98 First International Workshop on Computer Vision, Pattern Recognition and Image Processing, 1998.
- [8] Gonzales R.C, R.E. Woods Digital Image Processing Addison Wesley, 1993.
- [9] Koller, D., Weber, J., Malik, J. "Robust Multiple Car Tracking with Occlusion Reasoning." Proc. Third European Conference on Computer Vision, pp. 189-196. (1994).
- [10] Lee J.W, Kim M.S., Kweon I.S. "A Kalman-filter based Tracking Algorithm for an Object Moving in 3D 0-8186-7108-4/95 1995 IEEE (342-347).
- [11] Hyes Roth F., Waterman D.A., Lenat, D. B. Building expert systems. Addison Wesley, 1983.

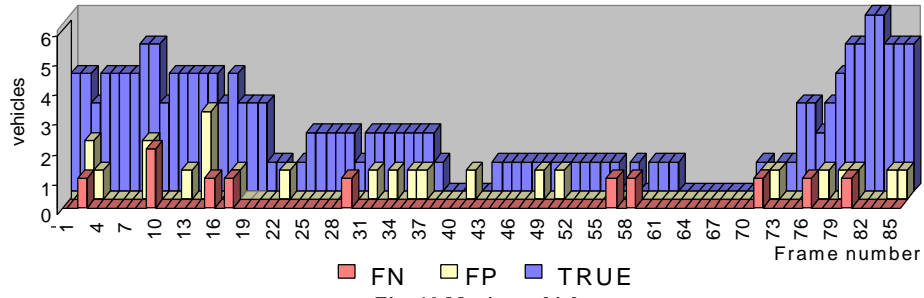


Fig. 10 Moving vehicles.

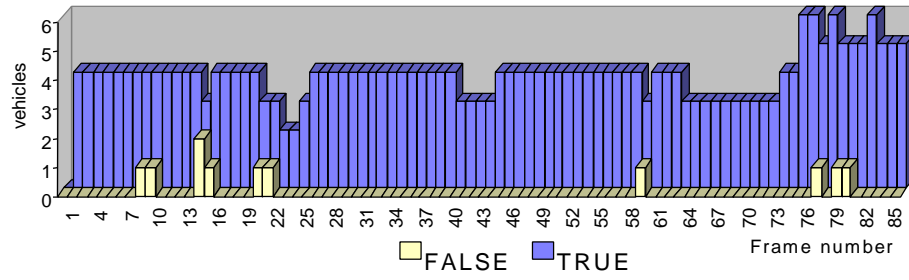


Fig. 11 Tracked vehicles.