HIGH-LEVEL TRAFFIC-VIOLATION DETECTION FOR EMBEDDED TRAFFIC ANALYSIS

Julien A. Vijverberg[1,2], Nick A.H.M. de Koning[1,2], Jungong Han[1], Peter H.N. de With[1], Dion Cornelissen[2]

(1) Eindhoven University of Technology, The Netherlands, (2) Prodrive B.V., The Netherlands

ABSTRACT

This paper presents the design of a *robust* and *real-time* traffic-violation detection system for cameras on intersections. We use background segmentation and a novel road-model to obtain the candidate traffic participants. A region-based tracking system, equipped with static occlusion-reasoning, tracks the positions of the objects in the scene. A computationally efficient camera model is defined which only requires three input parameters and enables the extraction of key object parameters like vehicle type and speed. Experiments have shown that an impressive average processing rate of 63-150 Hz is achieved, with high average correct road detection and object-type classification rates of 93-94% and event detection accuracy of 85%.

Index Terms—Background Subtraction, Machine Vision, Semantic Analysis, Traffic Information Systems

1. INTRODUCTION

Due to the growing number of cameras for surveillance and security purposes in society, the amount of processed video information (and partly recorded) is growing exponentially. To avoid excessively large databases with non-relevant data and make efficient use of the available bandwidth, it is desired to design cameras with content-analysis capabilities and networking facilities. A particularly interesting new application is automated traffic control for surveillance and detection of dangerous situations. This kind of high-level decision-making is only possible if the video is analyzed at the pixel, object and semantic level.

Significant research has been devoted to traffic analysis. Atev *et al.* [1] present a real-time method for detecting collisions on intersections. However, this system does not provide sufficient semantic-analysis results like the type of the vehicle, and they use a computationally expensive camera model. In [2], Lim *et al.* discussed stop-bar violation detection on intersections using areas assigned by an operator. The camera is calibrated on a fixed position and needs manual input, which decreases flexibility. Hu *et al.* [3] present an accident-prediction method using 3D object-models, which is computationally expensive and needs manual extraction of trajectories. The above examples are interesting but strongly limited to specific cases. A key paper for our work is [4], where Kumar *et al.* present a complete system, which can handle both highway and urban scenes. Unfortunately, this system has two major drawbacks: (1) suddenly revealed new background cannot be handled efficiently and (2) the camera model is complex and requires accurate and elaborate manual tuning.

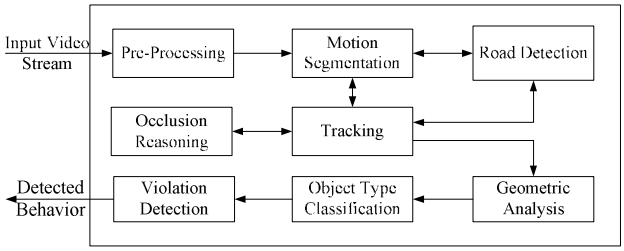

Fig. 1: Overview of the complete system.

This paper aims at the robust detection of traffic violations on urban intersections, while aiming at real-time embedded implementation and with a large flexibility in camera installation. The key features are vehicle tracking, classification and detection of predefined violations on urban intersections. It will be shown that by adding better models (such as road models) in the segmentation, the semantic level of analysis (violation detection) can be improved while enabling the real-time performance. Another attractive feature is the use of an elegant camera model, that does not require much input and is accurate enough to support behavior analysis in real-time.

2. ARCHITECTURE DESCRIPTION

The architecture of our system is illustrated in Fig. 1. Let us now briefly explain each module of the diagram.

The pre-processing module performs conventional image enhancement and control operations. The motion segmentation separates the objects from the background with assistance of the road model. To this end, a road-detection module is implemented which leads to the generation of a road model. The tracking block tracks all objects through consecutive frames. It uses a two-dimensional region-based blob tracking based on an adaptive double exponential smoothing predictor.The occlusion-reasoning module solves occlusions for objects disappearing behind static structures

and other objects. The geometric analysis module performs object-size measurements, which are used in the classification module to classify different object types. The violation detection uses event-raising boxes, which are user-defined areas in the image where specific event measurements are carried out to check for illegal object behavior.

The sequel of this paper concentrates on two novel aspects. First, the motion segmentation is enhanced with knowledge of a simple, six-parameter and yet accurate road model and information of automatically-learned occluding background structures. Second, a reduced pinhole-camera model for geometric analysis of traffic objects yields a high-performance classifier. As a result, we obtain one of the few complete violation detection systems achieving good results and real-time performance simultaneously.

3. ALGORITHM HIGHLIGHTS

3.1 Traffic Object Segmentation

Traffic-object detection is based on motion segmentation, which consists of background estimation and background subtraction. The quality of traffic-object segmentation heavily depends on the constructed background image. Ideally, a background estimator quickly adapts to changes, but is insensitive to the presence of real objects in the foreground. The structure of the algorithm is as follows.

1. Subtract background from input.
2. Perform foreground region tracking.
3. Selective update of background.
   a. Update if no object present using tracking results (This defines the initial binary exclusion mask).
   b. Check for correct background using road model (This defines the final binary exclusion mask).
   c. Integrate newly revealed background using road model.

The foreground regions of the tracked objects that have passed a series of consistency tests are appended to the exclusion mask. Non-traffic objects usually fail the consistency test and we propose to integrate such objects in the background to prevent *false* foreground regions. However, vehicles that just enter the scene and then stop for a relatively long period will not yet pass the consistency test, and will inevitably be confused for background. Earlier proposed systems cannot prevent this problem since they do not *know* whether the visible object in the background is correct. We solve this problem by obtaining this knowledge using an automatically constructed road model, because in traffic sequences large parts of the background consist of traffic roads.

When the road model classifies a background pixel as 'road' and the foreground pixel at the same position is 'not road', then it is a logical assumption that updating the background at that position will not give further improvements. Therefore, the road model is of great help in preventing the vanishing of real vehicles in the background. This is particularly beneficial in stopping zones of intersections.

Moreover, the road model also contributes to the correction of suddenly revealed background, since it provides information of what is expected to be visible in the background. If a new part of the background is revealed, it results in a false foreground region. The road model computes the number of pixels in such a region that is enclosed in the road model. If the majority of pixels in a region is classified as 'road', then it is considered as new background and is quickly incorporated in the background.

3.2 Road Detection

This algorithm aims at automatically estimating a model for the pixels in the background that belong to the traffic roads. The following model is used for road classification. The road model is derived in two algorithm steps.

1. A training set of road pixels is obtained. Since vehicles dominantly move *on* the road, the tracking system can construct the training set by keeping track of the positions where tracked objects frequently occur; the pixels at those positions in the background form the training set.
2. The variances and averages of the road model are computed. Initial values are obtained from the training set.
   a. A pixel $\boldsymbol{p}=(p_Y, p_U, p_V)$ is classified as 'road' when for each YUV-component i, with road average μ_i and standard deviation σ_i, the relation $|p_i - \mu_i| < \lambda_i\sigma_i$ holds, where the λ_i's are empirical thresholds.
   b. Post-processing on standard deviations σ_i is required to remove outliers from the training set and generalize for different road types. Since traffic roads typically contain little color, σ_U and σ_V are bounded to a maximum value, which reduces the effect of outliers. The computed initial σ_Y is usually large and therefore upper-limited to prevent a high false-positive classification rate. The same but opposite statement holds for the false-negative rate.

3.3 Occlusion Reasoning

We also present an outline of an algorithm capable of learning large occluding regions, such as trees. This allows model-based reasoning which can be used to improve the motion segmentation and tracking modules.

Occlusion is the most difficult problem in tracking systems. We make a distinction between static and dynamic occlusions regions. Existing systems mainly focus on

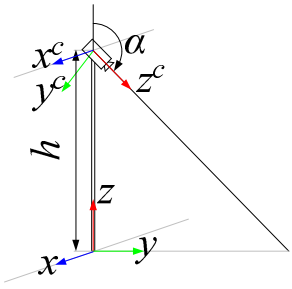

Fig. 2: External Camera model

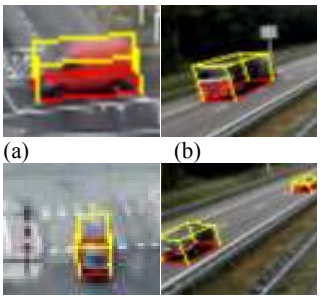

Fig 3. Results of 3D shoebox fitting.

dynamic (inter-object) occlusion, while *static* (object-scene structure) occlusion is overlooked, or only implicitly handled by heuristics. Our system handles static occlusion by obtaining the regions in the background that frequently occlude or *split* foreground objects, without explicit modeling. A vehicle splits e.g. when it drives behind a traffic light or traffic pole; this is called Thin-Structure Occlusion (TSO). TSO regions are obtained by looking for image positions where one side of the bounding box of a tracked object remains static, while the opposite side moves in the direction of the object motion. Since such objects disappear *behind* an object, there must be an edge visible in the background. We combine these two observations to extract TSO regions. Complete static occlusions are extracted by keeping track of image regions where objects frequently become fully occluded. The static occlusion regions are always in the same image positions, since we are using a static camera. The tracking system is improved by taking the locations of static occlusion regions into account.

3.4 Camera Model in the Geometric Analysis

In order to analyze a traffic scenario, we need to study the geometric relation among camera position, road, vehicle position, etc. The installer of the camera only has to define the height h and the angle α of the camera with respect to the ground plane, see Fig. 2. The focal length f is extracted from a Look-Up Table (LUT) relating the zoom factor and the focal length. The principal point (o_x , o_y) is set to the center of the image. Homogeneous image coordinates $\boldsymbol{p}'$ and ground-plane coordinates $\boldsymbol{p}$ are related by the homography matrix H_{xy} via $\boldsymbol{p}=H_{xy}\boldsymbol{p}'$ with H_{xy} being

$$H_{xy} = \begin{bmatrix} h & 0 & -o_x h \\ 0 & h\cos\alpha & fh\sin\alpha - o_y h\cos\alpha \\ 0 & \sin\alpha & -f\cos\alpha - o_y \sin\alpha \end{bmatrix}. \qquad (1)$$

The scaling factor in $\boldsymbol{p}$ only depends on the image coordinate y'. Thus, for any reasonable values of y', the inverse transformation to Euclidian coordinates can be implemented efficiently using a LUT, thereby saving numerous multiplications.

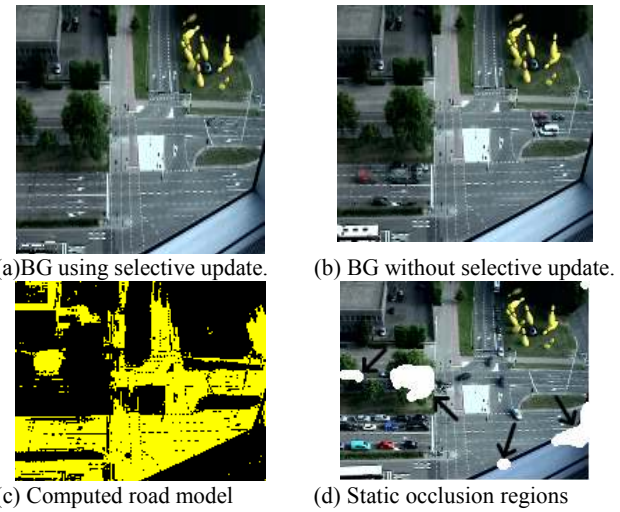

Fig. 4: Background (BG) result with (a) and without (b) selective updating using the road model of (c). Detected static occlusion regions are shown in (d).

The presented camera model is applied to distance measurements using three-dimensional shoeboxes. These are fitted around objects, using the vanishing points of the object's direction, similar to [1]. Visual results are illustrated in Fig. 3(a)-(d).

3.5 Object Type Classification

Classification is required by most semantic analysis applications in traffic. The feature vector $\boldsymbol{x}(n)$ measured at frame n is constituted from N_f=5 elements: (1) the vehicle- or object-length and (2) width (to discriminate e.g. cars from persons), (3) the object's maximum speed (km/h) for example to distinguish bicycles from motorcycles, (4) the bounding box filling degree e.g. for discriminating solid vehicles from "open" vehicles, and (5) the fraction of contour pixels for similar reasons as (4). Five object types are used (pedestrian, bicycle, motor, car, truck/bus). For each object type t, every feature f is modeled by its average $\mu_{t,f}$ and variance $\sigma_{t,f}^2$. A naive Bayesian classifier is used to determine the within-frame type likelihood $\varphi_t(n)$ for $\boldsymbol{x}(n)$ according to

$$\varphi_t(n) = \sum_{f=0}^{N_f-1} -\frac{1}{2} log(2\pi\sigma_{t,f}^2) + \sum_{f=0}^{N_f-1} -\frac{(x_f(n) - \mu_{t,f})^2}{2\sigma_{t,f}^2} . \qquad (2)$$

A simple update rule is employed, which only requires the accumulation of the overall likelihood $\varphi_{t,N}$ per type up to frame N. After applying this update rule, for each t, $\varphi_{t,N}$ is added to the currently highest likelihood and clipped to a certain minimum value to keep the results within boundaries.

4. PIXEL-LEVEL RESULTS

The result of the median-based background estimation enhanced with selective updating using the constructed road

model (see Fig. 4(c)) is shown in Fig. 4(a). Several groups of cars and a bus appear and they are waiting to enter the intersection. In the standard median-based approach, these traffic objects have vanished into the background, and thus tracking would not be possible. Consequently, it can be noticed that the estimated background in Fig. 4(a) is much better than the one in Fig. 4(b), which does not use the selective update algorithm. A second problem is that after background subtraction, the vanished vehicles lead to *false* non-traffic foreground. Also, due to the selective updating, Fig. 4(a) clearly shows significant improvement in the background image. The automatically constructed road model used to obtain this good result is illustrated in Fig. 4(c). Visual inspection of this road model shows that high performance can be achieved with our road model. An impressive average correct classification rate of 93% has been measured using manually generated ground-truth data for four different intersections. In Fig. 4(d), the static occlusion regions (for instance the trees) are correctly detected.

5. SEMANTIC-LEVEL RESULTS

The performance of the classifier has been tested with two traffic video sequences (10 and 7 min.) recorded at different intersections. The training data is extracted from a third sequence. From the results, we concluded that the classification into five object types, which satisfies most applications, leads to 94% correct decisions.

Finally, having defined all aspects, the overall system was tested to be good enough for making high-level semantic classifications. As an interesting example, we attempted to detect traffic violations. Experiments showed that one-way driving violations are measured quite easily, when the event raising boxes from the user are placed on suitable positions, leading to 100% correct violation detection. Another experiment showed that detecting illegal bus-lane driving is more difficult. We measured 10 false violation detections out of 13 detections and 2 false rejections out of 28 rejections, resulting in a correct classification rate of 70%. This lower rate is explained by initially classifying the wrong object type, which can be improved by simply logging the violation event and evaluating the object type after its disappearance. Table 1 summarizes our results.

The average frame rates for four sequences have been evaluated. The sequences were captured with 25-Hz frame rate and were MPEG-1 or MPEG-4 SP encoded. In order to obtain reliable measurements and a serious performance indication, more than 75 minutes of video data were evaluated. The traffic-analysis system was implemented on a P-IV 3.2-GHz computer. The minimum and maximum average frame rates are an impressive 63 Hz and 150 Hz, respectively, thereby offering real-time performance.

Table 1. Performance Summary

Performance Metric	Result
Road correctly classified	93%
Type correctly classified	94%
One-way driving correct	100%
Bus-lane violation correct	70 %

6. CONCLUSIONS

In this paper, we have presented a multi-level video analysis system for violation detection on urban intersections. The key of our system at the pixel-level is to very selectively update the background, which is achieved by the feedback of the tracking information and the incorporation of a novel road model. The road model helps in selectively supporting or rejecting foreground objects. This is especially useful on intersections, where traffic objects can be standing still for a long time. The road model can be implemented with high-performance giving on the average 93% correct classification rate. This attractive solution can be generalized to work for alternative background-estimation techniques as well.

The most important contribution of the semantic-level part is the efficient pinhole camera model. The traffic-object sizes are efficiently measured using three-dimensional shoebox models for the objects. This leads to a low-cost classifier with a low number of features and correct classification rates up to 94%. Example applications have shown that our system satisfies the requirements for semantic-level analysis.

The aim is to pursue real-time performance on an intelligent camera. The complete system has been implemented and achieves real-time performance (63-150 Hz processing rate), thereby enabling embedded realizations in intelligent cameras in the near future.

A possible enhancement of the system would be the inclusion of a shadow handler into the object-segmentation module, which would correct the image region of an object and split more falsely merged objects.

7. REFERENCES

[1] S. Atev, H. Arumugam, O. Massoud, R. Janardan, N. Papanikolopoulos, "A Vision-Based Approach to Collision Prediction at Traffic Intersections," *IEEE Trans. Intelligent Transportation Systems*, Vol. 6, no. 4, pp. 416-423, December 2005.

[2] D. Lim, S. Choi and J. Jun, "Automated Detection of All Kinds of Violations at A Street Intersection Using Real Time Individual Vehicle Tracking," *Fifth IEEE Southwest Symp. Image Analysis and Interpretation*, Santa Fe, USA, pp. 126-129, April 2002.

[3] W. Hu, X. Xiao, D. Xie, T. Tan, S. Maybank, "Traffic Accident Prediction Using 3-D Model-Based Vehicle Tracking," *IEEE Trans. Vehicular Technol.*, Vol. 53, no. 3, pp. 677-694, May 2004.

[4] P. Kumar, S. Ranganath, H. Weimin and Kuntal Sengupta, "Framework for Real-Time Behavior Interpretation From Traffic Video," *IEEE Trans. Intelligent Transportation Systems*, Vol. 6, no. 1, IEEE, pp. 43-53, March 2005.