ON-BOARD ROBUST VEHICLE DETECTION AND TRACKING USING ADAPTIVE QUALITY EVALUATION

Jon Arróspide, Luis Salgado, Marcos Nieto, and Fernando Jaureguizar

Grupo de Tratamiento de Imágenes - E. T. S. Ing. Telecomunicación
Universidad Politécnica de Madrid - Madrid (Spain)
{jal, lsa, mnd, fjn}@gti.ssr.upm.es

ABSTRACT

This paper presents a robust method for real-time vehicle detection and tracking in dynamic traffic environments. The proposed strategy aims to find a trade-off between the robustness shown by time-uncorrelated detection techniques and the speed-up obtained with tracking algorithms. It combines both advantages by continuously evaluating the quality of the tracking results along time and triggering new detections to restart the tracking process when quality falls behind a certain quality requirement. Robustness is also ensured within the tracking algorithm with an outlier rejection stage and the use of stochastic filtering. Several sequences from real traffic situations have been tested, obtaining highly accurate multiple vehicle detections.

Index Terms— Vehicle detection, tracking, RANSAC, optical flow, Kalman filtering.

1. INTRODUCTION

One of the most essential features regarding driver assistance systems (DAS) is the capability to detect other vehicles on the road. Monocular vision based systems are particularly interesting for their low cost and for the high-fidelity information they give about the driver environment. The on-board scenario imposes a bunch of intrinsic complexities, namely, the vehicle itself as well as the other vehicles may be moving, generally at different speeds. Thus, the background is constantly changing, and so is the appearance of the road in front of the vehicle.

The literature accounts for a whole range of approaches to face the problem of on-board vehicle detection, typically focused on looking for single features in the image, such as: shadows [1], symmetry [2][3], edges [4], etc. Other solutions [5] propose a combination of several features to make the detection more robust and reliable.

Nevertheless, vehicle detection strategies use independent analysis of each image in a sequence, which has a number of inherent limitations: e.g. (i) it provides no temporal coherence between vehicles detected in different frames; and (ii) individual detection performed in a set of images of a sequence is too costly for real-time applications.

Hence we have oriented our work to vehicle tracking, by using information from previous frames, in such a way that we spare processing time and are able to draw trajectories of the vehicles. In the literature, several approaches are presented to perform vehicle tracking in these dynamic environments [6][7]. Generally speaking, these strategies aim at finding the correspondence between every point in a sequence of images, namely, the optical flow, which describes the velocity of pixels in an image sequence. Nonetheless, as stated previously, the on-board scenario involves a dynamic environment where both elements (i.e., objects) and conditions change rapidly, and so finding a unique correspondence between points is not always achievable.

In this work, to overcome the limitations of independent detections and tackle the complexity of the scenario, a novel multiobject feature tracking strategy is proposed. It tracks specially selected points of the image (i.e. corners) based on computation of sparse optical flow. The tracking strategy includes a central outlier rejection stage, that ensures robustness of the tracker based on probabilistic techniques, and a Kalman filtering stage to smooth out the trajectories. Furthermore, as opposed to other works in which a unique initial detection is used [8] or periodically restarted [9], we evaluate the confidence of the tracking via a quality process, that adaptively triggers new detections when tracking is not acceptable.

2. SYSTEM OVERVIEW

Tracking is based on an initial detection of the vehicles calculated as in [5]. The tracking module delivers estimations of the new regions likely to contain each vehicle insofar as the quality measure associated to it is good. Whenever it comes below the accepted quality for one vehicle, a new detection is triggered restricted to an extended region to recompute vehicle location. The system operation is presented in Fig. 1.

The vehicle detection module delivers a set of l regions likely containing vehicles $\{\mathbf{r}^i\}_{i=1}^{l}$. Each region $\mathbf{r}^i$ is a vector defined by its upper-left point (x_0^i, y_0^i), its width w^i and its height h^i, i.e. $\mathbf{r}^i = (x_0^i, y_0^i, w^i, h^i)$.

The tracking process is performed over each vehicle based on a initial detection of a region at time k_0, i.e. $\mathbf{r}_{k_0}^i$ (for clarity, henceforth we will assume a specific object i and drop the index). The module consists of the following phases: feature extraction, feature matching, and region estimation. The feature extraction phase, in which corners within $\mathbf{r}_{k_0}$ are computed, is run only at the beginning of the tracking process, or every time a new detection is triggered and the tracking reset. At time $k = k_0 + 1$, if we assume a prior detection $\mathbf{r}_{k_0}$ at time k_0 over the image I_{k_0}, the feature matching phase finds the correspondences at time k of those corners in the current image I_k, denoted as $\{\mathbf{c}_{k_0}^i\}$, using optical flow computation and a robust outlier detection technique. Geometry of

This work has been partially supported by the European Commission 6th Framework Program under project IST-2004-027195 (I-WAY). This work has also been supported by the Ministerio de Ciencia e Innovacin of the Spanish Government under project TEC2007-67764 (SmartVision), and by the Comunidad de Madrid under project S-0505/TIC-0223 (Pro-Multidis).

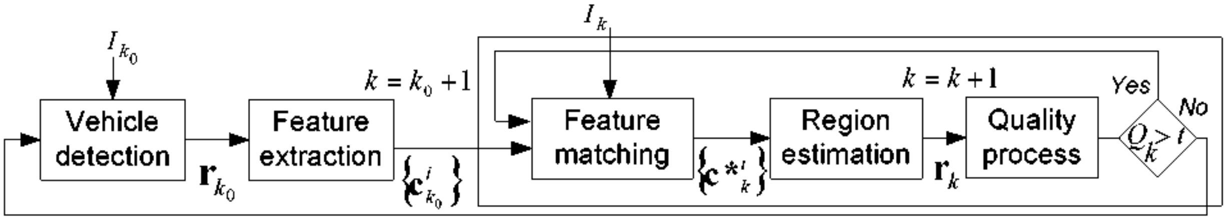

Fig. 1. Block diagram of the system

these features is used to estimate the new region $\mathbf{r}_k$ where the vehicle lies, using Kalman-based filtering. Feature matching and region estimation phases of the tracking are computed for every new image at time $k = k + 1$ using the previous region and its corners as input, until the quality of the estimation is not considered adequate, i.e. quality criteria is not fulfilled, and then a new detection is triggered for the vehicle.

3. VEHICLE DETECTION

Vehicle detection is performed in three phases [5]: segmentation, hypotheses generation and hypotheses verification. First, image segmentation is obtained by deriving a modified edge image that enhances vertical and horizontal edges. Regions of high edge density are located using a split and merge technique followed by a blob coloring technique. The result is a set of regions of interest $\{R_i\}$ that are likely to contain vehicles.

Next, the regions that may contain vehicles are fine-tuned and a set of more precise hypotheses, $\{H_i^j\}$, is generated that will be validated in the verification phase. In this phase, the lateral histograms of the modified edge image are computed and smoothed out using a $LPF = [1/4 \; \underline{1/2} \; 1/4]$. Peaks of the histograms correspond to the positions of the strongest vertical and horizontal edges, which are expected to match the borders of the vehicle. Pairs of peaks are then combined in both directions, restricted to geometrical constraints that accommodate the appearance of real vehicles, to form the final hypotheses.

Finally, a verification stage is performed to classify the resulting hypotheses as belonging to a vehicle class or not. A multidimensional feature vector composed of shadow, symmetry, and corner measures is used as explained in [5] to feed a minimum Mahalanobis distance classifier, that filters the hypotheses $\{H_i^j\}$ to yield a set of vehicle detections $\{\mathbf{r}^i\}_{i=1}^l$. Each hypothesis has an associated confidence measure p^i based on the relative distance of the feature vector to the vehicle class.

4. VEHICLE TRACKING

4.1. Feature extraction

Let us take k_0 as the triggering time of the tracking module for one vehicle. This implies that tracking is referred to the detection result $\mathbf{r}_{k_0}$ at time k_0. A set of base features has to be extracted for each detected vehicle in order to perform the correspondence matching in posterior phases. Corners are selected as good features to track due to their invariability to motion information and to environmental conditions. To obtain them, a Harris corner detector is used over the region $\mathbf{r}_{k_0}$. The result comprises a set of corners $\{\mathbf{c}^i_{k_0}\}_{i=1}^n$, that constitutes the basis for the tracking of the vehicle. The number of features n to detect must be subject to a trade-off between tracking time and reliability.

4.2. Feature matching

A robust feature tracking strategy is proposed based on the sparse optical flow computation between consecutive images, and a robust outlier rejection technique.

First, the Lukas-Kanade Tracking (LKT) algorithm is computed in the current frame to find correspondences of the corners $\{\mathbf{c}^i_{k-1}\}$ in region $\mathbf{r}_{k-1}$ of the previous frame. The algorithm might miss some correspondences due to corners exceeding the limits of the image or to the use of bad features, thus generally the results will consist of a smaller set of corners of length $m < n$, denoted $\{\mathbf{c}^i_k\}_{i=1}^m$.

However, these correspondences may be corrupted owing to noise, clutter, illumination or other effects. In order to ensure the robustness of the tracking, we propose a new outlier rejection technique based on RANSAC applied on a small data set. At a sufficient frame rate, the transformation between points in two consecutive images, can be modeled with an affine matrix A. Hence, for each point:

$$\hat{\mathbf{c}}^i_k = \mathrm{A}\mathbf{c}^i_{k-1} \tag{1}$$

For each possible combination of three corners from the data set $\{\mathbf{c}^i_k\}$ and their corresponding pairs in $\{\mathbf{c}^i_{k-1}\}$, the transformation matrix A^j is computed. Then the distance is calculated from each remaining point $\mathbf{c}^i_k$ to its corresponding point $\hat{\mathbf{c}}^i_k$ transformed with A^j. If this distance is larger than a preset threshold t the point is identified as an outlier. The transformation A^j that renders the lowest number of outliers is used as the best estimation, and the outliers associated to it are rejected. As a result, a smaller set of corners $\{\mathbf{c}^{*i}_k\}$ of length $m^* < m$ is obtained.

4.3. Region estimation

To estimate the movement of the vehicles, we make use of the fact that the perspective effect modifies only very slightly the structure of rigid objects in consecutive frames. This involves that the relations between corners of a vehicle are maintained from frame to frame, except presumably for a scale factor. For each vehicle, let us now define a measure ρ as:

$$\rho = \frac{1}{m^*} \sum_{i=1}^{m^*} d(\mathbf{c}^{*i}, \mathbf{v}) \tag{2}$$

where $d(\cdot)$ measures the Euclidean distance between a corner and the centroid of the corners $\mathbf{v}$:

$$\mathbf{v} = \frac{1}{m^*} \sum_{i=1}^{m^*} \mathbf{c}^{*i} \tag{3}$$

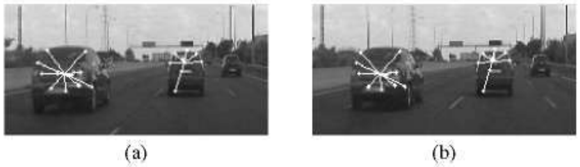

Fig. 2. Corner lattices at times k and $k+5$ of the same sequence.

At time instant k, the corners from the previous image that have a valid matching in the current image are expected to form a lattice similar to the one formed with current corners. The positions of the centroids of these lattices give us an estimation of the motion of the lattice. In turn, the parameter ρ measures the size of the lattice. A graphical example of the considerations above is given in Fig. 2, where the lattices of corners are shown for two captures of the same sequence separated by 5 frames. The structure of the lattice maintains its *shape* for both vehicles, while the size of the lattice corresponding to the vehicle in the left decreases slightly, as the vehicle moves away from the ego vehicle. The position of the centroid is also slightly shifted due to this movement. Hence, at time k we derive the parameters $\mathbf{v}_k$ and ρ_k using the corners $\{\mathbf{c}^{*i}_k\}$, and their dual measures $\mathbf{v}_{k-1}$ and ρ_{k-1} taking into account the correspondences of $\{\mathbf{c}^{*i}_k\}$ in the previous image. Then, to form the estimated regions $\mathbf{r}_k$, the position of the centroid is shifted from $\mathbf{v}_{k-1}$ to $\mathbf{v}_k$, maintaining the relative position of the centroid within the region. Finally, the region width and height are scaled by a factor $s = \rho_k/\rho_{k-1}$, that measures the change in size of the corner lattices.

A final trajectory smoothing step based on a Kalman filtering approach is performed to integrate estimations over time and obtain more reliable results. Namely, two linear position and motion Kalman filters are applied in cascade. The first filter has a state vector composed of the centroid coordinates (x and y) and their velocities (v_x and v_y), while the state vector of the second filter is composed by the value of the factor scale s and its velocity v_s. Observe that the filters are set in cascade due to the explicit dependence of the factor scale on the position of the centroid.

At the end of each tracking step some corners may be lost, as they can exceed the limits of the image, be classified as outliers or simply not be found with the LKT. To compensate it, and in order to keep the number of features as a constant n throughout all the tracking process, a Harris detector is applied over each region $\mathbf{r}_k$ if necessary and new corners are detected. This way, the size of the lattices formed by the corners is maintained, and thus the tracking process stabilized.

5. QUALITY MEASURE AND NEW DETECTION TRIGGERING

A quality measure is proposed to assess the tracking result, i.e. the estimated region. It is based on the computation of a symmetry value over the region $\mathbf{r}$ computed as in [3]:

$$f = \sum_{y=1}^{h} s(y) \tag{4}$$

where h is the height of the region $\mathbf{r}$, and $s(y)$ is calculated for each row of the region as

$$s(y) = \frac{\int_0^w {H_e}^2(x,y)dx - \int_0^w {H_o}^2(x,y)dx}{\int_0^w {H_e}^2(x,y)dx + \int_0^w {H_o}^2(x,y)dx} \tag{5}$$

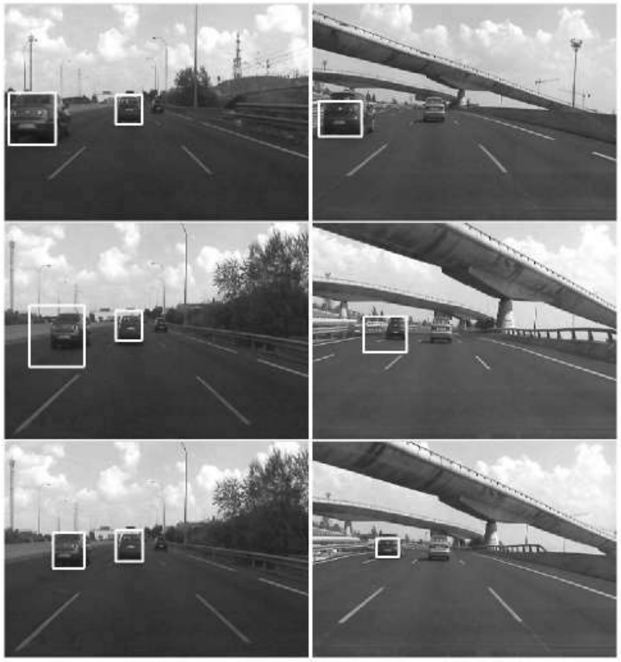

Fig. 3. Detection and Tracking examples for two sequences (a) and (b). First row shows detection of vehicles at time k_0. Second and third row show tracking of vehicles at time k_0+30 without and with outlier rejection module, respectively.

where w is the width of the region and

$$H_e(x,y) = \frac{I(x,y) + I(-x,y)}{2}$$
$$H_o(x,y) = \frac{I(x,y) - I(-x,y)}{2}$$

where $I(x,y)$ is the pixel value within the region. The indicator of the quality of the estimation at time k is given by:

$$q_k = \frac{f_k}{f_{k_0}} \tag{6}$$

where f_k is the symmetry value for the estimated region at time k and f_{k_0} is that computed for the detected region at time k_0. Therefore, q_k indicates the loss or gain in symmetry with respect to the initial reference value of the detection in k_0.

This parameter is computed for each vehicle i to obtain q_k^i, and it is combined with the confidence measure of each vehicle detection p^i at time k_0 (see Section 3) to give:

$$Q_k^i = q_k^i \cdot p_{k_0}^i \tag{7}$$

which takes into account not only the adequacy of the estimations but also the reliability of the initial vehicle detection.

At every instant k, if the Q_k^i goes below a threshold for any of the vehicles i, a new detection over an extended region is triggered for that vehicle at instant $k+1$.

6. RESULTS

Monocular grayscale sequences (360x288 pixels) have been recorded in a variety of very distinct situations (weather, road context, etc) in

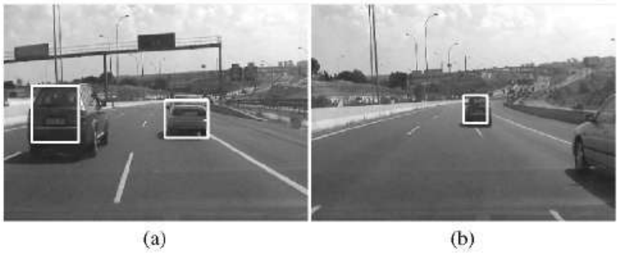

Fig. 4. Vehicle tracking based on a poor detection at time k_0 shown in (a). Image (b) shows the tracking of left vehicle at time $k_0 + 100$.

order to assess the performance of the algorithm in a broad scenario. Test results for tracking indicate that it provides highly reliable and precise estimations of vehicle position on the image upon a good initial detection. Fig. 3 presents tracking examples for two sequences with several vehicles: detected vehicles at time k_0 are shown in the upper row; tracking results at $k_0 + 30$ without and with the proposed outliers rejection strategy are shown below. As it can be observed, when a good detection is provided, the tracker is able to follow correctly the vehicles evolution. Observe that the use of the RANSAC outlier rejection module (third row in Fig. 3) boosts significantly the performance of the system, extends the reliability of the tracking, and consequently delays the triggering time for new detections. In addition, and remarkably enough, tracking example in Fig. 4 shows that the tracking system is actually capable of overcoming poor detections and adapting to the real tracks of the vehicles. In this figure, the left vehicle in (a) has not been accurately detected (its back part is only partially detected). Nevertheless, as shown in (b), after 100 images tracking process fits the estimated region very accurately to the vehicle. The tracking relies on the position of the centroid of the corners and its position relative to them, which are updated according to the vehicle movement, rather than on the initial bounding box provided by the detection, hence the estimated region eventually fitting the vehicle.

A symmetry measure over the estimated region is defined as an indicator of the goodness of the tracking. Graphs in Fig. 5 show the relative symmetry evolution of the vehicles in the previous examples. In Fig. 5(a), which corresponds to the left vehicle in Fig. 3(a), q_i is all time around 1, as corresponds to a good tracking. In turn, symmetry of the right vehicle in the same image, shown in Fig. 5(b), decreases notably when approaching the 30-th frame: the region containing the car is actually growing in its upper-left side, thus not being properly adjusted to the vehicle and losing symmetry. The quality module detects this situation, and triggers a new detection at frame $k_0 + 26$. This way, the bounding box of the vehicle is retrieved and the symmetry grows to higher values (dotted line in Fig. 5(b)), accepted by the quality process. Finally, as expected, average value of the symmetry measure grows well above 1 when the tracking system corrects a poor detection, as shown in Fig. 5(c), which corresponds to vehicle in Fig. 4.

7. CONCLUSIONS

In this work, we have presented a robust method for vehicle tracking, based on sparse optical flow computation and using a RANSAC-based outlier rejection technique, that filters wrong correspondences provided by optical flow.

To achieve tracking, we make use of the fact that the structure of the corners is maintained in rigid objects, hence the position of the centroid of the corners and its position relative to them being crucial to tracking. The proposed tracking system has been shown to yield very reliable results based on acceptable detections. Besides, it has been proved to be adaptive to the actual characteristics of the context, delaying or triggering new detections conveniently. Therefore, the system minimizes time-uncorrelated detections, with the consequent computational saving, and circumvents traditional limitations of tracking systems, combining both approaches to give a robust and flexible solution.

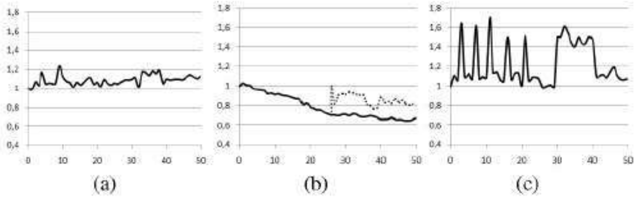

Fig. 5. Evolution of relative symmetry measure per frame: (a) for the left vehicle in Fig. 3(a), (b) for the right vehicle in the same image, and (c) for vehicle in Fig. 4. In (b), the dotted line indicates the symmetry measure rise after detection triggering at $k_0 + 26$.

8. REFERENCES

[1] T. K. ten Kate et al., "Mid-range and distant vehicle detection with a mobile camera," in *IEEE Proc. Intelligent Vehicle Symp.*, Parma, Italy, June 14-17, 2004, pp. 72–77.

[2] C. Hoffmann, T. Dang, and C. Stiller, "Vehicle detection fusing 2d visual features," in *IEEE Proc. Intelligent Vehicle Symp.*, Parma, Italy, June 14-17, 2004, pp. 280–285.

[3] T. Zielke, M. Brauckmann, and W. von Seelen, "Intensity and edge-based symmetry detection with an application to car-following," *CVGIP: Image Understanding*, vol. 58, pp. 177–190, 1993.

[4] Y. Chen, M. Das, and D. Bajpai, "Vehicle tracking and distance estimation based on multiple image features," in *IEEE Proc. 4th Canadian Conf. on Computer and Robot Vision*, Montreal, Canada, May 28-30, 2000, pp. 371–378.

[5] D. Alonso, L. Salgado, and M. Nieto, "Robust vehicle detection through multidimensional classification for on board video based systems," in *IEEE Proc. ICIP'07*, San Antonio, USA, Sept. 16-19, 2007, vol. 4, pp. 321–324.

[6] J. Shin et al., "Optical flow-based real-time object tracking using non-prior training active feature model," *ELSEVIER Real-Time Imaging*, vol. 11, pp. 204–218, June 2005.

[7] D. Serby and L. Van Gool E. K. Meier, "Probabilistic object tracking using multiple features," in *IEEE Proc. 17th Int. Conf. on Pattern Recognition.*, Istanbul, Turkey, Aug. 23-26, 2004, vol. 2, pp. 184–187.

[8] J. Van Leuven, M. B. Van Leeuwen, and F. C. A. Groen, "Real-time vehicle tracking in image sequences," in *IEEE Proc. Instrumentation and Measurement Conf.*, Budapest, Hungary, May 21-23, 2001, vol. 3, pp. 2049–2054.

[9] M. Lin and X. Xu, "Multiple vehicle visual tracking from a moving vehicle," in *IEEE Proc. ISDA'06*, Jinan, China, Oct. 06, vol. 2, pp. 373–378.