

Motion-based Obstacle Detection and Tracking for Car Driving Assistance

Gildas Lefaix, Éric Marchand, Patrick Bouthemy
IRISA - INRIA Rennes
Campus de Beaulieu, 35042 Rennes Cedex, France

Abstract

This paper is concerned with the detection and tracking of obstacles from a camera mounted on a vehicle with a view to driver assistance. To achieve this goal, we have designed a technique entirely based on image motion analysis. We perform the robust estimation of the dominant image motion assumed to be due to the camera motion. Then by considering the outliers to the estimated dominant motion, we can straightforwardly detect obstacles in order to assist car driving. We have added to the detection step a tracking module that also relies on a motion consistency criterion. Time-to-collision is then computed for each validated obstacle. We have thus developed an application-oriented solution which has proven accurate, reliable and efficient as demonstrated by experiments on numerous real situations.

1. Introduction

This paper is concerned with the detection and tracking of obstacles from a camera mounted on a vehicle with a view to driver assistance. This work is part of the Carsense European project [3]. The goal of this project is to develop a sensor system to deliver information on the car environment at low speed to assist low-speed driving. The combination of several sensors should improve object detection. Indeed various sensors are considered in the Carsense project: short and long range radar, laser sensor, and video cameras. Each sensor is nevertheless supposed to be independent. Extracted information from the different sensors will be fused at a further step in order to reduce false alarm rate and to handle possible missing data.

We are concerned with the image-based obstacle detection task. The aim is to detect obstacles located at less than 50 meters in front of the car. Such an obstacle could be a vehicle (car, truck), a motor-bike, a bicycle or a pedestrian crossing the car trajectory. The goal is to develop fast and efficient vision algorithms to detect obstacles on the road and to reconstruct their trajectories wrt. the camera, for all static or moving detected obstacle. To achieve this goal, we have designed a technique based on motion analysis as pro-

posed in [1, 2, 7, 4]. We first compute the dominant image motion assumed to be due to the car motion [4, 6]. It is represented by a 2D quadratic motion model which correspond to the projection of the rigid motion of a planar surface (here, the road). It is estimated by a robust multi-resolution technique [6]. Detection of obstacles can then be performed by considering that their apparent image motion is not conforming the computed dominant motion. Once detected the objects (or obstacles) are tracked using a motion-based process. The trajectories and the time-to-collision associated to each obstacle are then be computed and are supplied as input of the data-fusion module. In, e.g., [2] the expected image motion was computed using the vehicle odometry and was not estimated using image information. The main advantage of the presented algorithm is that it does not consider any information but the images acquired by the camera and that it allows fast computation on standard hardware.

The reminder of the paper is organized as follows. In Section 2 we outline the dominant image motion estimation method. Section 3 then describes the detection and tracking of the obstacles and the computation of the time-to-collision. In Section 4, we report experimental results related to our application.

2. Dominant motion estimation

The estimation of the 2D parametric motion model accounting the dominant image motion is achieved with a robust, multi-resolution, and incremental estimation method exploiting only the spatio-temporal derivatives of the intensity function [6].

Quadratic motion models. Since the near car environment free of obstacles is formed by the road which can be considered as a planar surface, the image dominant motion due to the car motion can be exactly represented by a 2D quadratic motion model involving eight free parameters. Let us denote $\Theta = (a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7)$, the velocity vector $\mathbf{W}_\Theta(P)$ at pixel $P = (u, v)$ corresponding to the quadratic motion is given by:

$$\mathbf{W}_\Theta(P) = \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} + \begin{bmatrix} a_2 & a_3 \\ a_4 & a_5 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} a_6 & a_7 & 0 \\ 0 & a_6 & a_7 \end{bmatrix} \begin{bmatrix} u^2 \\ uv \\ v^2 \end{bmatrix} \quad (1)$$

Dominant image motion estimation To estimate the dominant image motion between two successive images I_t and I_{t+1} , we use the gradient-based multiresolution robust estimation method described in [6]. To ensure robustness to the presence of independent motion, we minimize a M-estimator criterion with a hard-re-descending function. The constraint is given by the usual assumption of brightness constancy of a projected surface element over its 2D trajectory. Thus, the motion model estimation is defined as:

$$\hat{\Theta} = \underset{\Theta}{\operatorname{argmin}} E(\Theta) = \underset{\Theta}{\operatorname{argmin}} \sum_{P \in R(t)} \rho(\operatorname{DFD}_\Theta(P)) \quad (2)$$

$$\text{with } \operatorname{DFD}_\Theta(P) = I_{t+1}(P + \mathbf{W}_\Theta(P)) - I_t(P). \quad (3)$$

$\rho(x)$ is the Tukey's biweight function. The estimation support $R(t)$ can be the whole image. In practice, it will be restricted to a specific area of the image. The minimization is embedded in a multi-resolution framework and follows an incremental scheme. At each incremental step k , we can write: $\Theta = \hat{\Theta}_k + \Delta\Theta_k$, where $\hat{\Theta}_k$ is the current estimate of the parameter vector Θ . A linearization of $\operatorname{DFD}_\Theta(P)$ around $\hat{\Theta}_k$ is performed, leading to a residual quantity $r_{\Delta\Theta_k}(P)$ linear with respect to $\Delta\Theta_k$:

$$r_{\Delta\Theta_k}(P) = \nabla I_t(P + \mathbf{W}_{\hat{\Theta}_k}(P)) \cdot \mathbf{W}_{\Delta\Theta_k}(P) + I_{t+1}(P + \mathbf{W}_{\hat{\Theta}_k}(P)) - I_t(P)$$

where $\nabla I_t(P)$ denotes the spatial gradient of the intensity function. Then, we consider the minimization of the expression given by

$$E_a(\Delta\Theta_k) = \sum_P \rho(r_{\Delta\Theta_k}(P)). \quad (4)$$

This function is minimized using an Iterative-Reweighted-Least-Squares procedure. It means that the expression (4) is replaced by:

$$E_a(\Delta\Theta_k) = \frac{1}{2} \sum_P \omega(P) r_{\Delta\Theta_k}(P)^2, \quad (5)$$

and that we alternatively estimates $\Delta\Theta_k$ and update the weights $\omega(P)$ (whose initial value are 1).

This method allows us to get a robust and accurate estimation of the dominant image motion (i.e., background apparent motion) between two images.

3. Obstacles detection and tracking

3.1. Obstacles detection

The previous step supplies two sets of information:

- the motion parameters $\hat{\Theta}$ corresponding to the dominant motion;
- the map I_ω of the weights $\omega(P)$ used in (5) which account for the fact that pixel P is conforming or not to the computed dominant motion.

The maps I_ω will be used to detect the obstacles from the image sequences. Let us recall that the considered quadratic motion model is supposed to correspond to the apparent motion of the road surface. Therefore, each pixel that is not conform to this estimated motion can be considered as belonging to an obstacle, either static or moving (or to another part of the scene, if it does not lie on the road).

The outliers map I_ω is thresholded and mathematical morphology operators are applied in order to suppress noise. Pixels are then merged into group of pixels according to various criteria (distance or motion-based criterion). Concerning the distance criterion, two pixels or regions (group of pixels) are merged if they are connex or if the distance between these two groups is below a given threshold. Concerning the motion similarity criterion, two regions R_i and R_j are merged in a single region R_f if the motion within region R_i is consistent with the motion in R_f and if the motion in R_j is consistent with the motion in R_f . It is defined as:

$$C = C_{if} + C_{jf} \quad (6)$$

where C_{kf} reflects the motion similarity between regions $R_{k|k=i,j}$ and R_f :

$$C_{kf} = \frac{1}{\operatorname{Card}(R_k)} \sum_{P \in R_k} \left| d_{\hat{\Theta}_k}^u - d_{\hat{\Theta}_f}^u \right| + \left| d_{\hat{\Theta}_k}^v - d_{\hat{\Theta}_f}^v \right| \quad (7)$$

where $\mathbf{d}_\Theta = (d_\Theta^u, d_\Theta^v)$ denotes the displacement of pixel $P = (u, v)$ according to the motion model Θ , and $\hat{\Theta}_k$ designates the parameters of the motion model estimated within region R_k using the same method as the one described in Section 2. $\operatorname{Card}(R)$ is the number of pixels in region R .

3.2. Obstacle tracking

We then obtain for each obstacle (or, more precisely, for each area in the image that is not conforming to the dominant motion) a bounding area. In order to improve the efficiency of this approach and to limit the number of false alarms, these areas are tracked over successive frames.

A comparison between obstacle areas detected in two successive images is then necessary. Two tests both based on motion similarity have been investigated. The first one exploits the criterion defined in (6). The second, which assumes that the two regions overlap, and evaluates the motion consistency in the intersection of the two regions. It

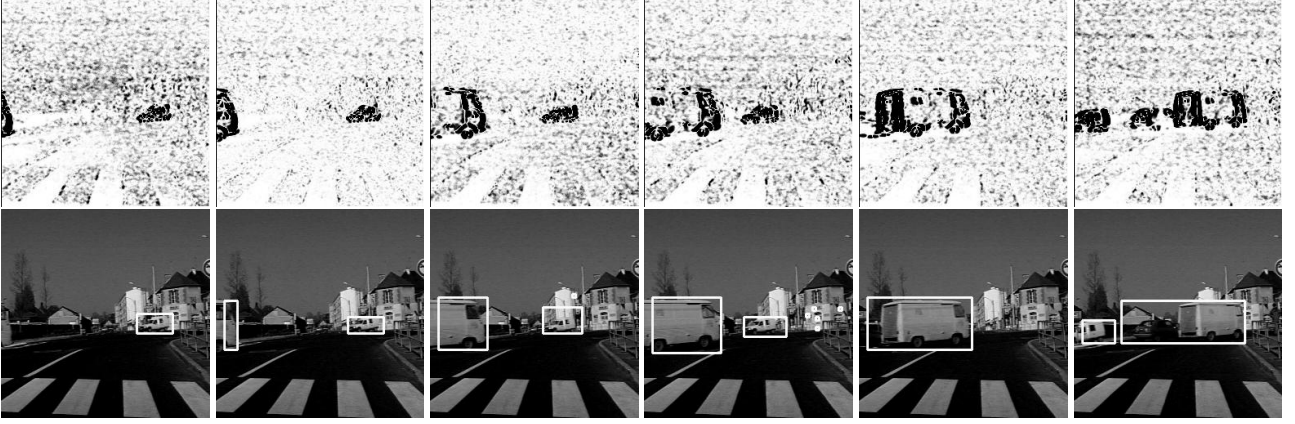


Figure 1. Obstacle detection from a motionless vehicle. The first and third rows display the outliers maps, the second and fourth rows contains the detected obstacles (see text for details)

can be expressed as follows:

$$C_{ij} = \frac{1}{\text{Card}(R_i \cap R_j)} \cdot \sum_{P \in R_i \cap R_j} \left| d_{\hat{\Theta}_j}^u - d_{\hat{\Theta}_i}^u \right| + \left| d_{\hat{\Theta}_j}^v - d_{\hat{\Theta}_i}^v \right| \quad (8)$$

This short term temporal tracking allow us to suppress false detections that could arise in a one-frame analysis. At this step we have a set of obstacle areas validated in the current frame and specified by their bounding rectangular boxes, and their position in the previous frame.

3.3. Time-To-Collision

For each detected obstacle, we can now compute the time-to-collision with the camera using the motion model computed in the bounding area. The time-to-collision is given by: $\tau_c(u, v) = -\frac{Z}{T_Z}$, where Z is the depth of the obstacle and T_Z the relative object velocity along the camera view axis. Assuming a planar object surface (at least a “shallow” object), we can write $Z = Z_0 + \gamma_1 X + \gamma_2 Y$ and τ_c can be rewritten as:

$$\tau_c(u, v) = -\frac{Z_0}{T_Z} \frac{1}{(1 - \gamma_1 x - \gamma_2 y)} \quad (9)$$

x and y are obtained assuming the knowledge of the camera intrinsic parameters: $x = (u - u_0)/p_x$ and $y = (v - v_0)/p_y$ ((u_0, v_0) are the coordinates of the camera principal point while p_x and p_y reflects the pixel size). γ_1, γ_2 and Z_0/T_Z can be computed from the affine parameters of the motion model $[a_1, a_2, a_3, a_4, a_5, a_6]$ defined in (1) computed within the bounding area for each obstacle at each instant. the relation have been derived in [5].

4. Results

The proposed method has been implemented on a PC running Linux. Each frame is processed in 0.4 s for

512×512 images and in 0.1 s for 256×256 images. We have tested our approach on numerous image sequences acquired in different contexts. We only report here four representative experiments.

In the first sequence, the car with the camera is motionless. Several vehicles are crossing the road in front of the camera (see Figure 1). Although this example is simple since it involves a static camera, it allows to validate the obstacles detection and tracking modules. The obstacles are successfully detected and tracked over the image sequence. In the fourth image of Figure 1 although the van and the car are close to each another they are correctly considered as two different vehicles since their motions are not homogeneous (according to the criterion of relation (6)). After they cross each other, they are again recovered (see the last image of Figure 1). In the same image the black car closely following the van is detected as a part of the same object. Indeed, it has the same motion features as the van.

In the second experiment, we consider a mobile camera following a van in an urban environment (see Figure 2). The van and the other vehicle (last image) are correctly detected and tracked over the image sequence. The time-to-collision was computed as described in Sub-section 3.3. Let us stress that the time-to-collision values plotted in Figure 2 have not been filtered over time (which could be easily implemented and could smooth the obtained plot). It proves that the time-to-collision computation is consistent. In experiments reported in Figure 3 and Figure 4 the car is moving within a more complex environment: a highway and a road. For efficiency concern, the dominant motion is not estimated in the whole image but only in an area delineated on the first image of Figure 3 (i.e., between the two dashed lines). Obstacles are again correctly detected and tracked even if a few false alarms may appear in some frames. Let us point out that objects lying close to the road as well as some road-

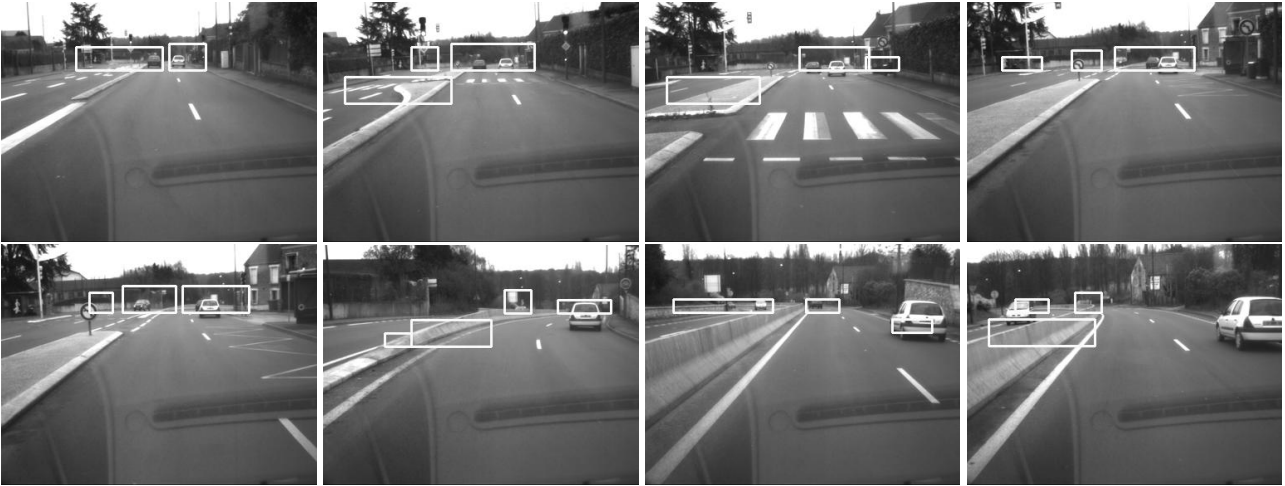


Figure 4. Detection of multiple (moving and static) obstacles in a road environment.

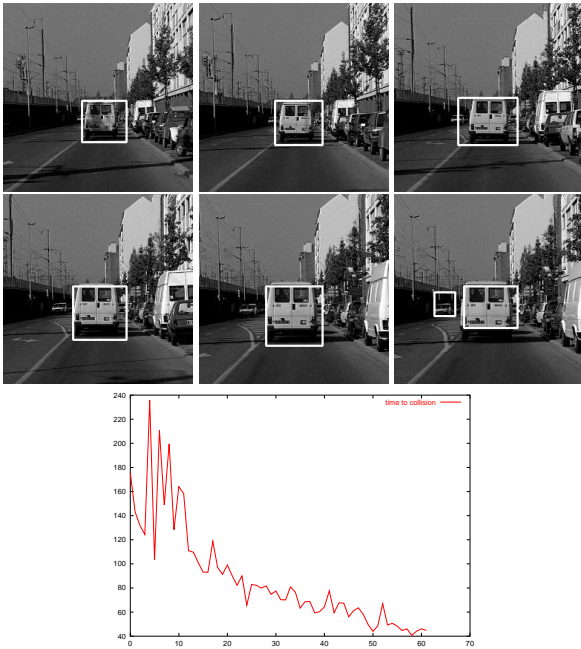


Figure 2. Obstacle detection and tracking in a urban environment. Plot corresponding to the computed time-to-collision values over time for the detected obstacle (white van).



Figure 3. Obstacle detection and tracking in a highway environment

signs are correctly detected.

5. Conclusion

We have presented an accurate and efficient method based on the robust estimation of the dominant image motion to detect obstacles in order to assist car driving. We have added to the detection step a tracking module that also relies on a motion consistency criterion. Time-to-collision is then computed for each validated obstacle. Our goal was to propose an algorithm that does not consider any other input but the images acquired by the camera. Furthermore we seek fast and robust computation on low-cost standard hardware. With respect to the considered application that is assistance at low-speed driving, quite satisfactory results on various real image sequences have validated our approach.

References

- [1] N. Ancona, T. Poggio. Optical flow from 1d correlation: Application to a simple time-to-crash detector. *IJCV*, 14(2):131–146, 1995.
- [2] W. Enkelmann. Obstacle detection by evaluation of optical flow field from image sequences. *Image an Vision Computing*, 9:160–168, 1991.
- [3] J. Langheim. Sensing of car environment at low speed driving. In *7th World Congress on Intelligent Transport Systems*, Torino, , Oct. 2000.
- [4] W. Krüger. Robust real-time ground plane motion compensation from a moving vehicle. *Machine Vision and App.*, 11(4):203–212, 1999.
- [5] F. Meyer, P. Boutheymy. Estimation of time-to-collision maps from first order motion models and normal flows. In *ICPR92*, pages 78–82, 1992.
- [6] J.-M. Odobez, P. Boutheymy. Robust multiresolution estimation of parametric motion models. *Journal of Visual Communication and Image Representation*, 6(4):348–365, December 1995.
- [7] J. Santos-Victor, G. Sandini. Uncalibrated obstacle detection using normal flow. *Machine Vision and Application*, 9(3):130–137, 1996.