An Embedded Multi-Modal System for Object Localization and Tracking

Sergio Alberto Rodriguez Florez, Vincent Fremont, Philippe Bonnifait, Véronique Cherfaoui

An Embedded Multi-Modal System for Object Localization and Tracking

Sergio A. Rodríguez F.[1,2], Vincent Frémont[1,2], Philippe Bonnifait[1,2], Véronique Cherfaoui[1,2]
[1]*Université de Technologie de Compiègne (UTC),* [2]*CNRS Heudiasyc UMR 6599, France*

***Abstract*—Reliable obstacle detection and localization is a key issue for driver assistance systems, particularly in urban environments. In this study a multi-modal perception approach is investigated, the objective being to enhance vehicle localization and dynamic object tracking in a world-centric map. A 3D ego-localization is achieved by merging information from a stereo vision system and data obtained from vehicle sensors. Mobile objects are detected using a multi-layer lidar that is also used to identify a constrained search space within the multiple target tracking process. Object localization and tracking is then performed in the fixed frame, which facilitates analysis and understanding of the scene. Experimental results using real world data are performed to evaluate the performance of the multi-modal system, and these are presented to show the effectiveness of the approach.**

***Index Terms*—Multi-modal perception, visual odometry, object tracking, dynamic map, intelligent vehicles.**

I. Introduction

Advanced Driver Assistance Systems (ADAS) can improve road safety through their obstacle-detection and collision-avoidance features. In this context, knowing the position and the speed of surrounding mobile objects is crucial.

The literature includes a number of different approaches to tackling the object localization and tracking problem. SLAM-like approaches can be used to characterize the static part of the environment and to simultaneously detect moving objects [1]. In [2], [3] and [4], two different stereo vision strategies are proposed to obtain a 3D dynamic map by combining 2D intensity and 3D depth information. A lidar alone can be used to estimate the ego-motion and to detect mobile objects via a dense 3D grid-map approach [5]. Alternatively, [6] presents a lidar based approach which performs mobile object tracking within a stochastic recursive Bayasien framework. In [7] and [8], real-time sensor-referenced approaches (i.e. ego-localization is not considered) are presented using multi-sensor systems, thus showing the complementarity of lidar and vision systems in automotive applications. Recently, Weigel et al. [9] have investigated a complete multi-modal object localization and tracking system using a grid based space representation. Their proposed system performs object detection based on lidar data, CAN bus sensors are used for motion compensation and a camera is employed to determine the height of objects.

For the purposes of object localization and tracking, a world-centric approach presents interesting properties for cooperative ADAS applications once the ego-localization has been accurately estimated. One well-known drawback of this approach is an unbound error propagation over long distances if it is based on odometry [10]. However, this strategy facilitates the motion model equations of the mobile objects and increases the accuracy of the tracking system over short distances, as discussed in [11].

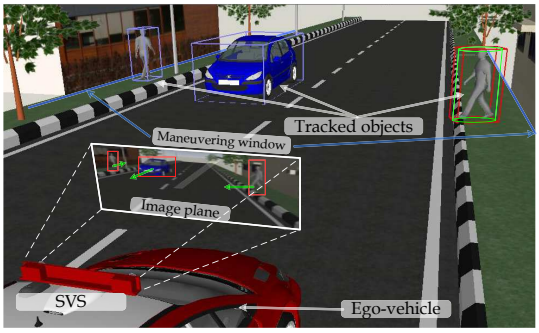


Figure 1: A dynamic map corresponds to a list containing the kinematic states of the tracked objects and the vehicle dynamics in the 3D scene

Ego-localization can be achieved using environment perception and vehicle sensor data. GPS is an affordable system that provides 3D positioning. Unfortunately, GPS performance can be significantly reduced in urban environments because of multi-paths and satellite outages. Odometry is a complementary solution which can provide good positioning when the relative motion is integrated over short distances. In this way, Stereo Vision Systems (SVS), often used for detection and recognition tasks, can also be useful for localization and navigation applications [12]. Several studies have already looked at how odometry using vision sensors (called in the following *visual odometry*) can provide precise positioning of a mobile platform in complex dynamic environments. Scaramuzza et al. [13] have proposed a stable feature tracking performing in the large field-of-view provided by an omnidirectional camera. Such a method dramatically reduces association errors, but it still remains a monocular approach subject to provide estimates up to a scale factor. In [14], full 6-

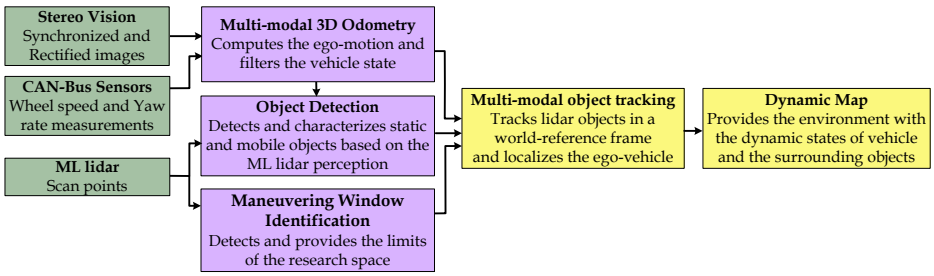


Figure 2: Multi-Modal Strategy: perception functions and their interactions for the object localization and tracking

DOF (degrees of freedom) motion is estimated using a robust filtering estimator, subsequently enhanced by a feature-classification scheme in [15]. In contrast, other methods make use of voting strategies [16] and decoupled rotation/translation estimates [17] with constrained motion models. In a previous work [18] we proposed and implemented a 6-DOF real-time visual odometry strategy based on a sparse optical flow, and on multiple view rigidity constraints introduced in [19].

Object tracking from a mobile platform remains an active research domain for ADAS. Urban environments are characterized by complex and dynamic conditions: moving (whether holonomic or non-holonomic) and static objects, and very different infrastructures. Object representation [20], [21], association methods [22], motion model and tracking strategies [23] are key points requiring particular attention.

We present here a multi-modal system able to provide a 3D local perception of the vehicle environment in a world-centric frame (see Fig. 1). The environment is composed of static and moving objects, and a maneuvering window is defined in front of the vehicle. The contribution of this work is estimating the dynamics of the surrounding objects (location and speed) based on different sensing modalities in order to build a *dynamic* map comprising a list of states of tracked objects and the changing vehicle dynamics in the 3D scene.

The embedded multi-sensor system employed uses proprioceptive sensors (i.e. wheel speed sensors and a yaw rate gyro) and two exteroceptive sensors: a Multi-Layer lidar (denoted ML lidar) and a Stereo Vision System (denoted SVS).

The overall strategy is described in Fig. 2. First, the vehicle ego-localization is estimated by merging CAN-bus (Controller-Area Network) information with visual odometry. The ML lidar then provides a 3D perception of the scene structure used to identify a maneuvering window. Following this, objects situated within the zone of interest are localized in the fixed-reference frame by compensating for the motion of the ego-vehicle. Finally, objects are tracked in this world frame and the resulting information can be used by an ADAS to estimate possible collisions.

This paper addresses 3D ego-localization, as well as object localization and tracking. Section II gives a detailed description of the embedded multi-sensor system setup. The different multi-modal functions are next presented and discussed. Section III is devoted to multi-modal 3D odometry using vision and proprioceptive sensors. Object localization and tracking are studied in Section IV. Finally, Section V presents conclusions and discusses future work.

II. Multi-Modal Perception System

Let us consider a vehicle equipped with an ML lidar, a yaw rate gyro, wheel speed sensors (WSS) accessible through a CAN-bus gateway, and a stereo vision system (SVS). These instruments supply the asynchronous inputs into our perception system.

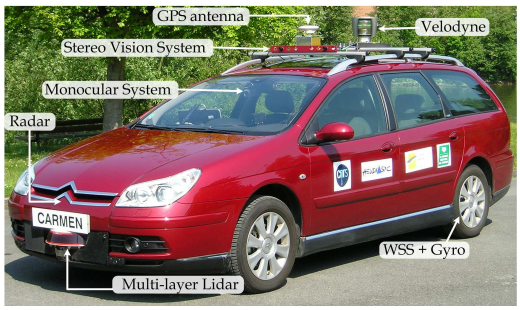


Figure 3: The experimental vehicle and its different perception capabilities. This study makes use of a stereo vision system, the IBEO Alasca XT and wheel speed and gyroscopic sensors.

Our vehicle is equipped with an IBEO Alasca XT lidar which provides a sparse perception of the 3D environment by the means of 4 crossed-scan-planes covering $150°$ horizontally in a $200\,\mathrm{m}$ range. Perception of the surroundings is complemented by a $47\,\mathrm{cm}$-baseline Videre SVS installed on the roof of the vehicle and covering a $45°$ field of view. Vehicle data is acquired through a CAN-bus gateway giving access to the speed of the rear-wheels and the yaw rate measurements.

A. Coordinate systems

Each of the perception devices references the acquired data with respect to its own frame. Lidar measurements are reported in a Cartesian frame, denoted $\mathcal{L}$ (X-Front,

Y-Left and Z-Up) and refreshed at a rate of 15 Hz. SVS data is referenced to the vision frame, $\mathcal{S}$ (X-Right, Y-Down and Z-Front) and updated at 30 FPS. Vehicle measurements (WSS and Gyro) are reported in a frame located at the center of the rear-axis, denoted $\mathcal{R}$ (X-Right, Y-Down and Z-Front).

As stated above, our work aims at generating a dynamic map using different sensing modalities. To this end, the transformations between the different sensor frames have to be determined, as illustrated in Fig. 5. The geometrical transformation between the ML lidar and the SVS is denoted as ${}^{\mathcal{S}}[\mathbf{q},\,\mathbf{t}]_{\mathcal{L}}$ and is determined via an extrinsic calibration [24]. The transformation between the SVS and the World frame is denoted ${}^{\mathcal{W}}[\mathbf{q},\,\mathbf{t}]_{\mathcal{S}(t)}$, and changes over time since we are dealing with a mobile platform. This transformation is computed using a real-time odometry method presented later in this paper.

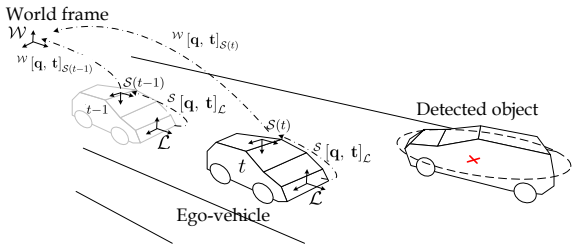


Figure 5: Coordinate systems

III. Multi-modal 3D Odometry

Multi-modal 3D odometry corresponds here to the 3D pose estimation of the ego-vehicle as a function of time with respect to a fixed initial frame. Odometry methods using stereo vision systems can provide very precise 3D pose estimations based on quadrifocal constraints, as presented by Comport et al. [19]. However, visual odometry may require significant computation time, since it makes use of a dense image warping technique. Real-time execution consequently calls for optimization and adequate computational resources.

In order to achieve a good trade-off between precision and execution time, we estimate the 3D vehicle ego-localization using visual odometry in addition to the odometry information obtained from the CAN-bus sensor measurements. A global overview of this alternative localization technique is illustrated in Fig. 4.

A. Visual odometry aided with CAN-bus sensors

The ego-motion of the vehicle is defined by an elementary transformation (rotation-translation composition, 6 DOF) over a certain time interval. This estimate is represented by an axis-angle rotation and a translation vector, ${}^{\mathcal{S}(t-1)}[\Delta\boldsymbol{\omega},\,\Delta\mathbf{v}]_{\mathcal{S}(t)}$. We first make an estimate of planar motion using the CAN-bus sensors over a time interval Δt (see *Motion initialization* in Fig. 4). This initial guess is then fed into a 3D visual motion estimation algorithm that iteratively refines the solution (see *Robust function* in Fig. 4).

Let $\mathcal{R}(t)$ be the center of the rear-wheel axis defined at time t. If the sampling frequency of the gyro and the WSS is sufficiently high (around 40 Hz), the wheel speed is almost constant and the planar ego-motion can be approximated by a circular arc. As illustrated in Fig. 6, the planar ego-motion of the vehicle is modeled as follows:

$$\Delta\boldsymbol{\omega}_0 = \begin{bmatrix} 0 \\ \Delta\theta \\ 0 \end{bmatrix} \quad \Delta\mathbf{v}_0 = \begin{bmatrix} \Delta s \cdot sin(\Delta\theta/2) \\ 0 \\ \Delta s \cdot cos(\Delta\theta/2) \end{bmatrix} \quad (1)$$

where $\Delta\theta$ is the angle obtained by integrating the yaw rate, Δs is the integrated rear-wheel odometry in meters, $\Delta\boldsymbol{\omega}_0$ is a vector representing the axis-angle rotation of the vehicle motion and $\Delta\mathbf{v}_0$ is a vector representing the estimated displacement of the rear-wheel axis center.

The estimated motion ${}^{\mathcal{R}(t-1)}[\Delta\boldsymbol{\omega}_0,\,\Delta\mathbf{v}_0]_{\mathcal{R}(t)}$ is then considered as a near estimate of ${}^{\mathcal{S}(t-1)}[\Delta\boldsymbol{\omega}_0,\,\Delta\mathbf{v}_0]_{\mathcal{S}(t)}$.

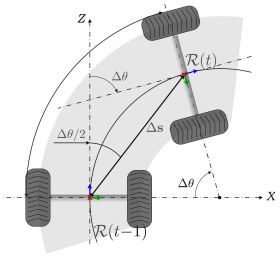


Figure 6: Yaw rate-WSS for planar odometry estimation

Using successive stereo image pairs, denoted $\mathbf{I}^*$, $\mathbf{I}'^*$ and $\mathbf{I}''$, $\mathbf{I}'''$ (see Fig. 7), we obtain a set of tracked stereo feature points, $\widetilde{\mathbf{p}}''$, $\widetilde{\mathbf{p}}'''$, and their corresponding optical flow constituting the image motion. For this purpose a set of stereo feature points, $\mathbf{p}^*$, $\mathbf{p}'^*$, is extracted using Harris features associated with a ZNCC (Zero-mean Normal Cross Correlation) correlation criterion and image constraints (disparity and epipolar constraints). Here, $\mathbf{p}_i^*$ and $\mathbf{p}_i'^*$ are defined as the projection of an observed 3D point $\mathbf{P}$. The stereo features, $\mathbf{p}^*$, $\mathbf{p}'^*$, are tracked over time using the Lucas-Kanade method [25], thus defining the tracked stereo feature points set $\widetilde{\mathbf{p}}''$, $\widetilde{\mathbf{p}}'''$.

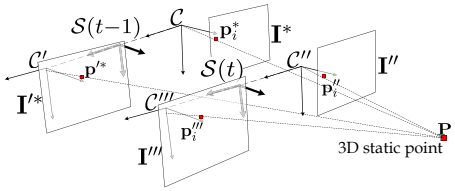


Figure 7: Quadrifocal warping principle

A stereo feature can be predicted after a 3D motion of the vision system using a warping function [19] based

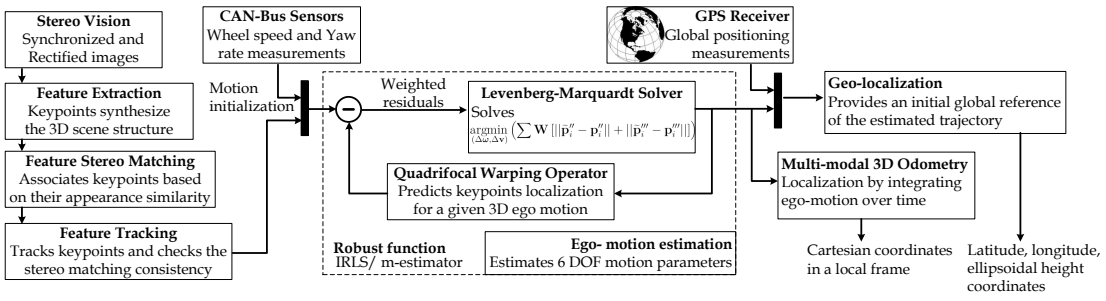


Figure 4: Localization system overview

on geometrical constraints entailed by by the stereo configuration and by the assumption that the scene is static. The main idea is to predict the set, $\mathbf{p}''$, $\mathbf{p}'''$, as a function of the set $\mathbf{p}^*$, $\mathbf{p}'^*$, of stereo features at time t and the vehicle's motion incorporated in the trifocal tensors.

As stated in [26], the simplified constraint parameters comprise the tensors linking the camera triplets $(\mathcal{C}, \mathcal{C}', \mathcal{C}'')$ and $(\mathcal{C}', \mathcal{C}, \mathcal{C}''')$, denoted respectively $\mathcal{T}_i^{jk}$, $\mathcal{T}_l^{mn}$, and the fundamental matrix $\mathbf{F}_{34}$ linking the cameras $(\mathcal{C}'', \mathcal{C}''')$, which is equivalent to $\mathbf{F}_{12}$ for the cameras $(\mathcal{C}, \mathcal{C}')$ (see Fig. 7). This parametrization remains robust to camera modeling errors and depends on the unknown motion parameters only, ${}^{\mathcal{S}(t-1)}\left[\Delta\boldsymbol{\omega},\ \Delta\mathbf{v}\right]_{\mathcal{S}(t)}$. The stereo warping operator is thus given by:

$$\begin{bmatrix} p''^k \\ p'''^n \end{bmatrix} = \begin{bmatrix} p^{*i}\, l'_j\, \mathcal{T}_i^{jk} \\ p'^{*l}\, l_m\, \mathcal{T}_l^{mn} \end{bmatrix} \tag{2}$$

where l'_j and l_m are respectively the covariant representations of the left and right image lines passing through the image points p^{*i} and p'^{*l}, and perpendicular to the epipolar line. The image points p^{*i}, p'^{*l}, p''^k, p'''^n are the contravariant representations of $\mathbf{p}^*$, $\mathbf{p}'^*$, $\mathbf{p}''$, $\mathbf{p}'''$.

Thus far, the visual odometry scheme is valid under the rigid scene assumption. If this assumption is not satisfied, the estimated ego-motion will be biased or completely incorrect. However, such an assumption is not realistic for intelligent vehicle applications, since the SVS will be operating in environments characterized by mobile objects, complex backgrounds and occlusions. The relaxation of the rigid scene assumption can be made through the use of a robust estimator which will also prove useful in dealing with other sources of error like image noise, stereo matching and tracking drifts. It should be remarked that scene can no longer be assumed static in the presence of such errors.

To this end, a robust iterative non-linear minimization is performed on the following criterion:

$$\underset{{}^{\mathcal{S}(t-1)}[\Delta\boldsymbol{\omega},\ \Delta\mathbf{v}]_{\mathcal{S}(t)}}{\arg\min} \left(\sum_{i=1}^{s} \mathbf{W} \left[||\widetilde{\mathbf{p}}_i'' - \mathbf{p}_i''||_2 + ||\widetilde{\mathbf{p}}_i''' - \mathbf{p}_i'''||_2 \right] \right) \tag{3}$$

where $||\cdot||$ represents the L2-Norm operator, s is the number of tracked stereo feature pairs, and $\mathbf{W}$ is the weighting matrix estimated by an M-estimator function [27] updated using an Iterative Re-weighted Least Squares algorithm (IRLS).
This robust minimization converges into a solution by rejecting the feature points that are generated principally by mobile objects. A consistent solution is obtained if at least 50% of stereo feature points correspond to static objects (i.e. the environment). The criterion of Eq. 3 is minimized by using the Levenberg-Marquard Algorithm (LM) in an IRLS loop [27]. The convergence speed of the LM algorithm is increased using the planar ego-motion ${}^{\mathcal{R}(t-1)}\left[\Delta\boldsymbol{\omega}_0,\ \Delta\mathbf{v}_0\right]_{\mathcal{R}(t)}$ from the CAN-bus sensors, this information providing a close initialization guess and then helping to reduce the number of iteration cycles.

After convergence, the 3D localization of the vehicle with respect to the initial position $\mathcal{S}(t=0)$ is estimated using the following state evolution equations.

Let $\mathcal{S}(t) = {}^{\mathcal{S}}\left[\mathbf{q}(t),\ \mathbf{p}(t)\right]$ be the 3D vehicle position at time t with $\mathbf{q}(t) = \left[q_0\ q_1\ q_2\ q_3\right]^T$ and $\mathbf{p}(t) = \left[p_0\ p_1\ p_2\right]^T$ representing the attitude as a unit quaternion and the vehicle position in meters. $\mathcal{S}(t)$ can be computed as follows:

$$\mathbf{q}(t) = {}^{\mathcal{S}(0)}\mathbf{q}_{\mathcal{S}(t-1)} \star^{\mathcal{S}(t-1)} \mathbf{q}(\Delta)_{\mathcal{S}(t)} \tag{4}$$

$$\underline{\mathbf{p}}(t) = {}^{\mathcal{S}(0)}\mathbf{q}_{\mathcal{S}(t-1)} \star^{\mathcal{S}(t-1)} \underline{\boldsymbol{\Delta}\mathbf{v}}_{\mathcal{S}(t)} \star^{\mathcal{S}(0)} \bar{\mathbf{q}}_{\mathcal{S}(-1)} + \ldots \ldots {}^{\mathcal{S}(0)}\underline{\mathbf{t}}_{\mathcal{S}(t-1)} \tag{5}$$

where $\mathbf{q}(\Delta\boldsymbol{\omega})$ is the unit quaternion corresponding to the estimated rotation of the vehicle ego-motion ${}^{\mathcal{S}(t-1)}\left[\Delta\boldsymbol{\omega},\ \Delta\mathbf{v}\right]_{\mathcal{S}(t)}$, $\star$ denotes the multiplication quaternion operator and $\bar{\mathbf{q}} = \left[-q_0\ q_1\ q_2\ q_3\right]^T$ is the conjugated unit quaternion of $\mathbf{q}$. Underlined vectors (e.g. $\underline{\mathbf{p}}$) denote expanded forms (i.e. $\underline{\mathbf{p}} = [0,\ \mathbf{p}]^T$) for the use of the quaternion multiplication.

B. Experimental Real-time 3D Ego-Localization Results

A data set was acquired in an urban environment featuring low-rise buildings, trees and moving objects (i.e. pedestrians and vehicles). During the experiment, a landmark on the road was employed as a start/stop ground truth of the vehicle's trajectory and the vehicle's speed was limited to 30 Km⁄h. The vehicle trajectory was

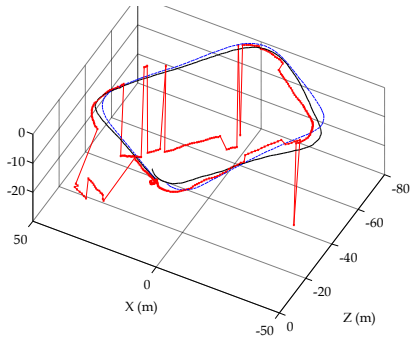
a closed loop featuring pedestrians and vehicles. Low-textured scenes (e.g. rural environments and parking lots) were not considered in this study.

The 3D ego-localization function is implemented in C/C++. The 3D trajectory is reconstructed in real time and is obtained by integrating the ego-motion estimations. Fig. 8 illustrates one of the tests performed. It consists of a 270 m-clockwise loop (i.e. 90 s video sequence duration).

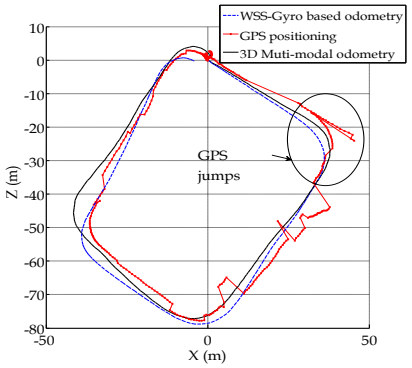
These results show that the cooperative strategy helps to cope with errors in the CAN-sensor based odometry, mainly due to wheel slippage and gyrometer bias drift. This technique also improved the visual odometry performance in adverse conditions (e.g. high rotational speed in 90° turns and roundabouts). These improvements were obtained thanks to the planar motion initialization which reduces the risk of converging into a local minimum ego-motion solution. It also improves outlier rejection and reduces the minimization iteration cycles. The 3D ego-localization system performs quite well in situations where GPS cannot provide a precise position (see the GPS jumps shown in the upper part of Fig. 8).

The bird's eye view of the estimated trajectories in Fig. 8 shows the gaps at the loop closing point. The 2D trajectory obtained using the WSS-Gyro based odometry (the blue curve) achieves an acceptable drift representing 1.84% of the total traveled distance (i.e. 4.17 m for a closed loop).

The total drift of the multi-modal strategy, computed as the Euclidean distance between the starting and the final trajectory position, was 1.9% of the total traveled distance. However, the drift of the planar trajectory projection of the multi-modal algorithm corresponds to 0.58% only, which represents an error 3 times lower than that obtained from the WSS-Gyro based odometry.



(a) 3D view of the reconstructed trajectory



(b) Bird's eye view of the closed-loop trajectory

Figure 8: 3D Reconstructed Trajectory. The circled region evidences GPS jumps, while visual odometry provides smooth estimates.

In Fig. 9, the multi-modal 3D odometry results were geo-localized. The aerial view of the trajectory is compared to a classical GPS localization. These results clearly show the improvements in vehicle localization resulting from the proposed approach.

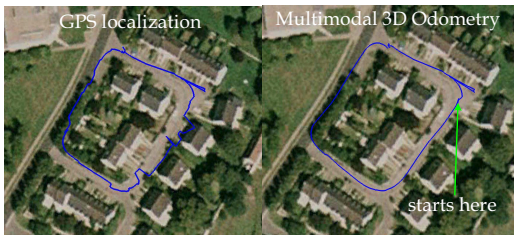


Figure 9: Aerial view of the GPS and multi-modal 3D odometry trajectory projections

IV. Object Localization and Tracking

The goal of this stage is to estimate the planar trajectory of a set of objects as they move in the 3D scene using a multi-modal approach (i.e. vision, lidar and WSS-Gyro sensing modalities). Object tracking also helps to maintain the temporal coherence of the dynamic map and to provide information about the objects' speeds and sizes.

The proposed multi-modal strategy starts by detecting objects using a lidar-based technique only. Objects are

then tracked using a Kalman filter algorithm with motion constraints in a characterized zone of interest. These constraints correspond to assumptions made in order to simplify the object tracking problem.

The tracking strategy is presented in four parts: object detection; maneuvering window identification; track prediction; object-track association and updating. At the end of this section, experimental results are reported to show the effectiveness of the approach.

A. *Maneuvering Window*

Urban environments are complex, dynamic and completely variable. Large numbers of static and mobile objects may considerably reduce the computational efficiency of an object tracker. A key issue here is identifying objects which constitute potential maneuvering targets so that computational resources can be mainly devoted to tracks requiring particular attention. Moreover, many incorrect observation-to-track pairings can be avoided. In the literature, the concept of a *maneuvering window* has been widely studied in military MTT[1] applications [28]. This concept has been recently proposed for pedestrian detection applications [29], where this zone is estimated based on the ego-vehicle state (typically speed and steering angle).

For the present study, the research space is restricted to a part of the 3D scene which is identified based only on prior knowledge of the environment (e.g. ground plane / urban canyon). Restricting the research space in this fashion provides the tracking algorithm with the ability to efficiently focus the computational resources on a maneuvering space, where collisions might be predicted at appropriate reaction times. Fig. 10 provides an example of the maneuvering window concept in an urban environment.

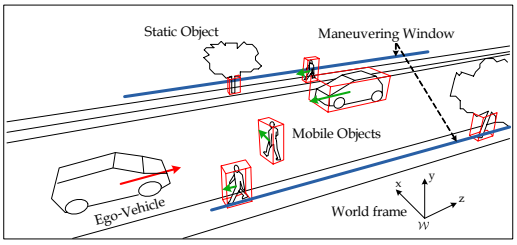


Figure 10: Maneuvering window

The method is based on the detection of maxima in lidar scan histograms. A first lidar data filtering is performed not only to improve the maneuvering window detection, but also to significantly decrease scene clustering issues. This filtering consists in detecting the 3D lidar data corresponding to the road surface. Detection makes use of the characteristic pattern observed when the two lower layers intersect the road plane at different angles, as illustrated in Fig. 11. The second lidar layer can be predicted from the first lidar layer and from the geometrical constraints. The measurements belonging to the road plane are detected and excluded from further processes, when the Euclidean error between the predicted and the measured layer falls below a predefined threshold. The pitch angle of the ML-lidar, denoted ρ, is estimated by a temporal filtering and is updated using the detected road impacts. The parameters h (ML-lidar height) and γ (inter-layer angle) are considered known.

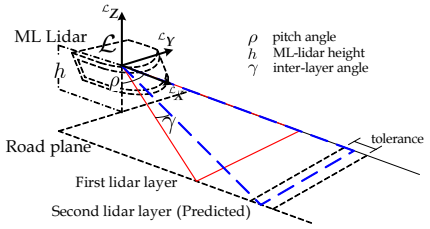
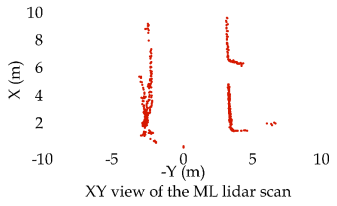
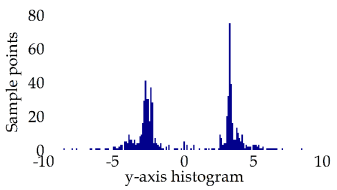


Figure 11: Characteristic pattern observed when two layers of the ML lidar intersect the road plane. This geometrical constraint is used to identify which scan points lie in the road plane so as to exclude them from further processes.

The maneuvering window is characterized by two local limits in the x-axis direction of the lidar frame. As illustrated in Fig. 12, a 4-layer data scan is projected onto the ${}^{\mathcal{L}}XY$ plane (see Fig. 12a) and provides an easy-to-exploit histogram ${}^{\mathcal{L}}Y$ axis (see Fig. 12b). Objects like security barriers, walls and parked vehicles efficiently reduce the maneuvering window. The detected limits are finally filtered using a fixed-gain Luenberger observer in order to reduce the oscillations produced by important pitch changes situations [29].



(a) Bird's eye view of laser scan



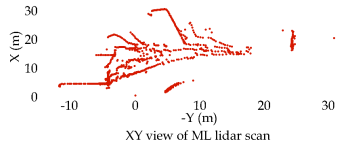
(b) Lidar scan projection into the y-axis

Figure 12: Maneuvering window identification using a y-axis histogram of the ML lidar scan points

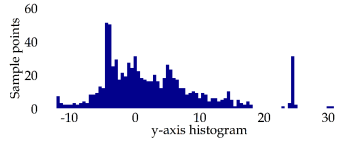
In turns and roundabout scenarios histogram peaks may fade out, as illustrated in Fig. 13. In this case, the

[1] *Multiple Target Tracking*

predicted localization of the maneuvering window limits are not associated with new observations, the updating stage of the fixed-gain filter does not take place and the last maneuvering window estimation is retained.



(a) Bird's eye view of a lidar scan in a turn situation



(b) Turning lidar scan projected into the y-axis

Figure 13: ML lidar y-axis histogram peaks in turns may fade out. To deal with this, the detected limits are filtered using a fixed-gain Luenberger observer.

B. Object Detection

The object detection function provides a list of 2D objects at each scan cycle. The detection involves a 3D point clustering stage which can be efficiently implemented using maximal Euclidean inter-distance [30]. Predefined geometrical features [21] are an alternative, but they require prior knowledge of objects. The output objects are characterized by their planar position in the lidar frame $\mathcal{L}$, their dimension (i.e. bounding circle) and detection confidence indicators [30].

The list of 2D object positions provided by the ML lidar at time t are transformed into the camera frame $\mathcal{S}$ and finally reported in a world frame (i.e. local dynamic map), $\mathcal{W}$, by compensating for the vehicle's motion (see section III-A).

For instance, let ${}^{\mathcal{L}}\widetilde{\mathbf{y}}(t) = [x \;\; y \;\; 0]^T$ be the coordinates of a detected object at time t in the lidar frame, as illustrated in Fig. 14. Its corresponding localization in $\mathcal{W}$ can be computed as follows, by composing two rigid transformations:

$$
\begin{aligned}
{}^{\mathcal{S}(t)}\underline{\widetilde{\mathbf{y}}}(t) &= \left({}^{\mathcal{S}}\mathbf{q}_{\mathcal{L}} \star^{\mathcal{L}} \underline{\widetilde{\mathbf{y}}}(t) \star^{\mathcal{S}} \bar{\mathbf{q}}_{\mathcal{L}}\right) +^{\mathcal{S}} \underline{\mathbf{t}}_{\mathcal{L}} \qquad (6)\\
{}^{\mathcal{W}}\underline{\widetilde{\mathbf{y}}}(t) &= \left({}^{\mathcal{W}}\mathbf{q}_{\mathcal{S}(t)} \star^{\mathcal{S}(t)} \underline{\widetilde{\mathbf{y}}}(t) \star^{\mathcal{W}} \bar{\mathbf{q}}_{\mathcal{S}(t)}\right) +^{\mathcal{W}} \underline{\mathbf{t}}_{\mathcal{S}(t)} \qquad (7)
\end{aligned}
$$

where ${}^{\mathcal{S}(t)}\widetilde{\mathbf{y}}(t)$ and ${}^{\mathcal{W}}\widetilde{\mathbf{y}}(t)$ are respectively the coordinates of the detected object in the SVS and in the world frame at time t. It will be recalled that the transformation denoted as ${}^{\mathcal{S}}[\mathbf{q},\, \mathbf{t}]_{\mathcal{L}}$ refers to the geometrical transformation between the SVS and the ML lidar frames, determined via an extrinsic calibration procedure.

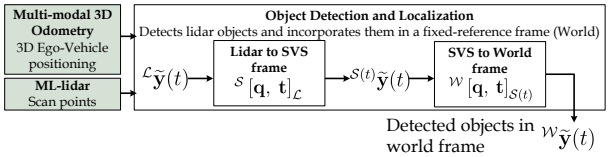


Figure 14: The detected objects are localized in the world frame by compensating for vehicle motion.

Assuming that the objects included in the maneuvering window locally follow a planar motion, the object state can be efficiently reduced to the ${}^{\mathcal{W}}XZ$ components of its Cartesian coordinates in the world frame. By abuse of notation the object state is represented as ${}^{\mathcal{W}}\widetilde{\mathbf{y}}(t)_{(x,z)}$. The uncertainty of the lidar object localization is modeled through a zero-mean, white, Gaussian measurement noise with a covariance denoted $\mathbf{N}$.

The track state is denoted by ${}^{\mathcal{W}}\mathbf{y}(t)$ and is composed of the ${}^{\mathcal{W}}XZ$ plane coordinates, $[x(t),\, z(t]$ in meters, and the planar velocity $[v_x(t),\, v_z(t)]$ in $m \cdot s^{-1}$ stacked in a 4D vector as follows:

$$
{}^{\mathcal{W}}\mathbf{y}(t) = \left[\; x(t) \quad z(t) \quad v_x(t) \quad v_z(t) \;\right]^T \qquad (8)
$$

This notation is intended to represent the knowledge of ${}^{\mathcal{W}}\mathbf{y}(t)$ obtained by combining all the information acquired up to time t. In addition to the state parameters, other attributes are handled in parallel, including the object size in meters (i.e. bounding circle radius) and the creation and update time-stamps in µs.

C. Track Prediction

The studied tracking system detects and tracks objects usually present in urban environments, such as pedestrians, cyclists and vehicles. Since their movements are linked to a fixed reference with a sampling frequency assumed to be sufficiently high, these movements can be taken as locally linear with a constant speed during a sample interval, Δt. This model is then given by:

$$
\mathbf{A}_T = \begin{bmatrix} \mathbb{I}_{2\times 2} & \Delta t \cdot \mathbb{I}_{2\times 2} \\ \mathbf{0}_{2\times 2} & \mathbb{I}_{2\times 2} \end{bmatrix} \qquad (9)
$$

with $\mathbb{I}_{2\times 2}$ as the 2×2 identity matrix.

Accordingly, the prediction of the track state is given by the following evolution equations:

$$
\begin{cases} {}^{\mathcal{W}}\mathbf{y}(t) = \mathbf{A}_T \cdot {}^{\mathcal{W}}\mathbf{y}(t-1) + \boldsymbol{\alpha}_T(t) \\ {}^{\mathcal{W}}\mathbf{y}(t)_{(x,z)} = \mathbf{C}_T \cdot {}^{\mathcal{W}}\mathbf{y}(t) + \boldsymbol{\beta}_T(t) \end{cases} \qquad (10)
$$

with $\mathbf{C}_T = \left[\; \mathbb{I}_{2\times 2} \quad \mathbf{0}_{2\times 2} \;\right]$.

${}^{\mathcal{W}}\mathbf{y}(t|t-1)$ is the predicted state of the tracked object, ${}^{\mathcal{W}}\mathbf{y}(t)_{(x,z)}$ is the observed track location, and $\mathbf{C}_T$ is the observation matrix and Δt is the sampling time period, which is not constant. $\boldsymbol{\alpha}_T(t)$ and $\boldsymbol{\beta}_T(t)$ are additive errors considered as white zero-mean Gaussian noises.

D. Object-Track Association and Updating

After an initial object detection sampling, a new set of tracks is created and all track parameters are set up. When the next incoming object arrives (i.e. ML-lidar sampling), iterative tracking actions are performed, starting with a state prediction of all the tracks at the current sampling time. The association process consists in selecting and assigning the closest new detected object to the predicted track position. It uses a nearest neighbor criterion based on the Generalized Statistical Distance metric $d(.)$ which maximizes the object-track association probability [28]:

$$d\left({}^{\mathcal{W}}\mathbf{y}(t|t-1),{}^{\mathcal{W}}\widetilde{\mathbf{y}}(t)_{(x,z)}\right) = \boldsymbol{\mu}(t)^{T}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}(t) + \ln|\boldsymbol{\Sigma}| \tag{11}$$

with $\boldsymbol{\mu}(t) = \mathbf{C}_{T}\cdot{}^{\mathcal{W}}\mathbf{y}(t|t-1) - {}^{\mathcal{W}}\widetilde{\mathbf{y}}(t)_{(x,z)}$, the state innovation, $\boldsymbol{\Sigma} = \mathbf{M}(t|t-1)+\mathbf{N}$, the residual covariance matrix and $\mathbf{M}$ is the covariance matrix of the track.

In order to cope with object occlusions, the non-associated tracks are kept for a fixed time interval (for example, 2 s). The non-associated objects in the maneuvering window generate new tracks until the algorithm reaches a maximum number of tracked objects.

The tracks' states and their corresponding covariances are updated using the information provided by the associated lidar object positions and classical Kalman filter equations.

Nearest neighbor (NN) methods imply single-hypothesis association approaches. These methods address ambiguous assignments by retaining the best available observation (i.e. the closest one). More sophisticated methods can address this issue, one example being Joint Probabilistic Data Association (JPDA) which is an extension, for the MTT problem, of the Probabilistic Data Association (PDA) in Single Target Tracking (STT) [28].

We have limited the scope of this study to the test of a simple assignment method that provides acceptable results through the use of a maneuvering window to filter and considerably reduce the number of candidates (which may be large in urban conditions.)

E. Experimental Results

3D ego-localization, maneuvering window detection and object-detection are real-time functions whose results have been logged (see the function scheme in Fig. 2). Fig. 15 shows the mean output frequencies of the 3D localization function and the ML lidar-based functions (namely maneuvering window and object detection). It will be observed that the convergence time of the 3D ego-localization function is not constant because it depends on the vehicle motion and the varying complexity of the scene.

The object tracking function was implemented with MATLAB. The reported results were obtained in an offline process. The inputs to the tracking algorithm were the logged results of the 3D ego-localization and the ML lidar-based functions.

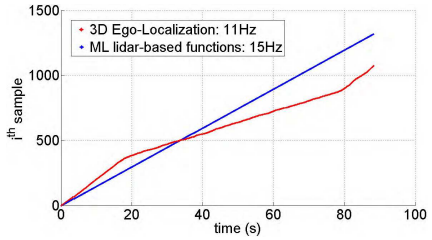


Figure 15: Real-Time Output Frequency of 3D Ego-Localization and ML lidar based Functions

Fig. 16 illustrates the XZ view of the reconstructed maneuvering window in the local map. For this reconstruction we use the 3D ego-localization of the vehicle and the ML lidar-vision extrinsic parameter results presented in the previous sections. It is important to highlight that at the beginning of the test sequence (i.e. initial position (0,0) in the XZ view) the vehicle remains static, which shows how the boundaries of the window converge. These results constitute a very interesting feature which can be linked to a GIS (Geographic Information System) for map-matching applications.

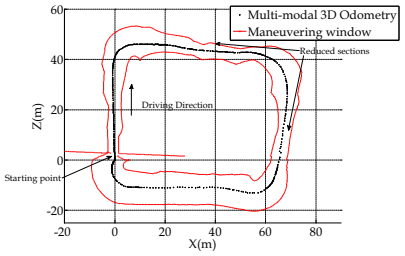


Figure 16: Reconstruction of the maneuvering window $\left({}^{\mathcal{W}}XZ \text{ plane view}\right)$. Notice the convergence of the filter after initialization. Reduced sections are due to parked vehicles

Focusing now on the kinematic state estimation of the tracked objects, Fig. 17 illustrates a zoomed area of the dynamic map. This zoomed area shows some state samples of a tracked vehicle that is accelerating and its corresponding track re-projection on the left SVS camera image.

Since our implementation provides a precise time reference (in µs), a reliable indicator of the correctness of the obtained results can be obtained by projecting predictions about tracked objects onto acquired images. Images here are only used as a means of identifying ground truth.

At the bottom of the figure, the size of the track is represented by its bounding circle, in red, and its center as a red triangle. The corresponding image track projections (3D red boxes) and their speed vectors

(green line) are also illustrated in the upper part of the figure. The projected bounding box encloses the 3D cylinder of the track.

Additionally, Fig. 17 shows how the size of a detected track changes as the object surface is impacted by the ML lidar. This fact sometimes entails perturbations in the speed estimation, since large changes in size give rise to a spurious motion in the track centroid. This problem has been addressed in [20], [29], where the spurious centroid displacement and the occlusions were dealt with using a model based vehicle tracking. Looking at the image projection of the track speed vector, however, one can see that the multi-modal system performs quite well. It is worth mentioning that objects are tracked even if they go out of the SVS field-of-view.

Fig. 18 shows another section of the dynamic map. Estimating the speed of a pedestrian is a challenge, since pedestrians' movements can be unpredictable. However, a linear motion at constant speed has shown to be extremely pertinent.

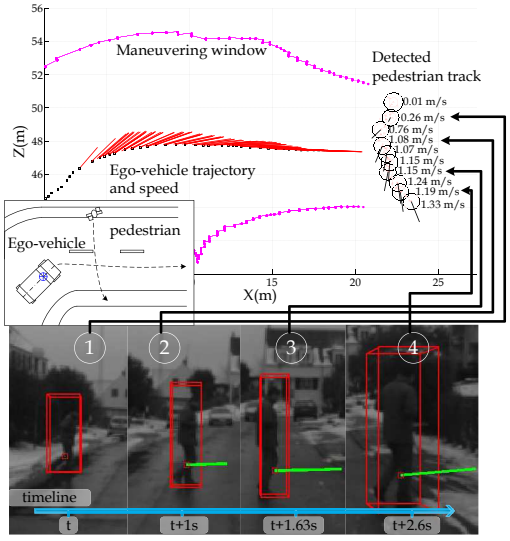


Figure 18: Trajectory of a pedestrian (XZ plane view)

The reconstructed trajectory illustrated in Fig. 18 corresponds to the pedestrian's real trajectory. Indeed, the pedestrian is following a semi-circular path while crossing the road and the estimated speed is in accordance with the real one of the pedestrian who was motionless at the beginning and reaches a usual human walking speed [31]. The filter converges in less than 0.4s (i.e. five laser scans in average), giving information on the kinematic characteristics of the tracks that are difficult to estimate using only a lidar.

In Fig. 19 a wheelchair pedestrian moving slow is successfully tracked even if the ego-vehicle has a high rotational speed because of a 90°-turn. Thanks to the accurate lidar measurements and the good estimation of the vehicle displacement, the tracking system is able to correctly recover the motion of the mobile. Indeed, the wheelchair goes firstly on the sidewalk, then follows it and finally turns right.

V. Conclusion and Future Work

An embedded multi-modal system for object localization and tracking has been proposed and experimentally validated.

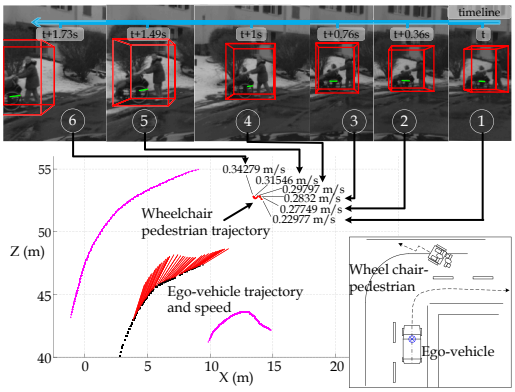


Figure 19: Wheelchair pedestrian trajectory observed from the moving vehicle in a 90° turn

The presented approach provides a 3D *dynamic* map of the vehicle's surroundings. The method merges sensor data to achieve a robust and precise 3D ego-localization that is crucial in compensating for the displacement of the ego-vehicle in the tracking process. This function is combined with a lidar-based object tracking focused in a maneuvering window, providing objects' trajectories and speeds as they move in space. The results obtained facilitate analysis and understanding of the scene, and can be used for ADAS (e.g. collision detection and avoidance).

The conducted investigations have led us to identify interesting clues which worth pursuing in a future research. A first perspective concerns the real-time implementation of the presented tracking approach which could be possible thanks to its low complexity. A further improvement of the complete perception architecture would allow the system to deal with dense traffic and crowded urban scenarios. Another perspective aims at improving trust in object detection and tracking through the use of a visual confirmation features which would allow to perform a statistical evaluation of the system performance. Finally, anyone interested in testing a new approach on the real dataset can address a request to the authors.

VI. Acknowledgments

The authors would like to thank Fadi Fayad for the implementation of the ML lidar functions and Gerald Dherbomez and Thierry Monglon for their experimental support

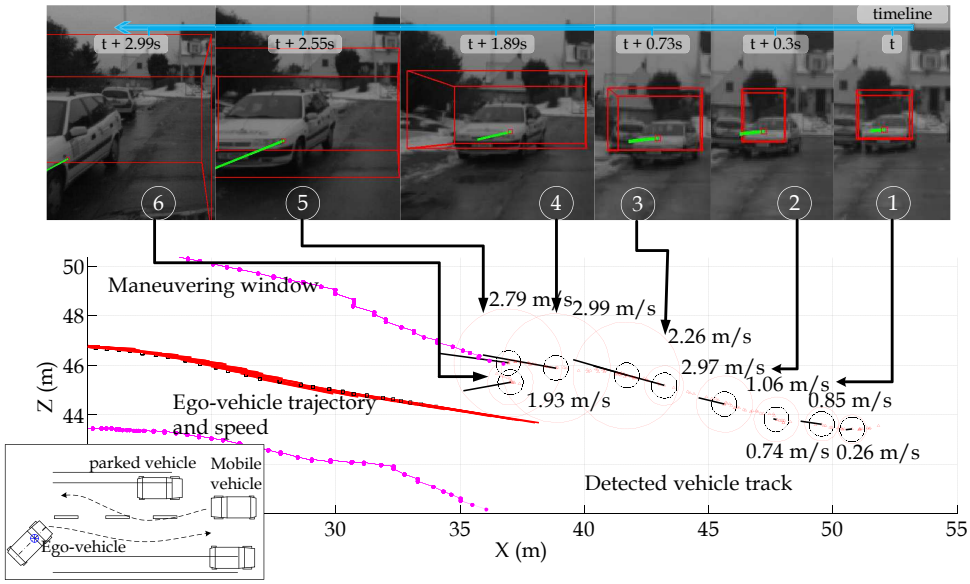


Figure 17: Trajectory of a tracked vehicle represented by red boxes. Track speed vector in green. The detected track size is illustrated by red circles. Black circles represents track covariance uncertainty.

REFERENCES

[1] C.-C. Wang, C. Thorpe, M. Herbert, S. Thrun, and H. Durrant-Whyte, "Simultaneous localization, mapping and moving object tracking," *International Journal of Robotics Research*, vol. 26, pp. 889–916, 2007.

[2] A. Barth and U. Franke, "Estimating the driving state of oncoming vehicles from a moving platform using stereo vision," *IEEE Trans. on Intelligent Transportation Systems*, vol. 10, pp. 560–571, 2009.

[3] B. Leibe, N. Cronelis, K. Cornelis, and L. V. Gool, "Dynamic 3d scene analysis from a moving vehicle," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR*, vol. 1, 2007.

[4] S. Nedevschi, S. Bota, and C. Tomiuc, "Stereo-based pedestrian detection for collision-avoidance applications," *IEEE Trans. on In Intelligent Transportation Systems*, vol. 10, pp. 380–391, 2009.

[5] T. Miyasaka, Y. Ohama, and Y. Ninomiya, "Ego-motion estimation and moving object tracking using multi-layer lidar," *IEEE Intelligent Vehicles Symposium*, vol. 1, pp. 151–156, 2009.

[6] S. Gidel, P. Checchin, C. Blanc, T. Chateau, and L. Trassoudaine, "Pedestrian detection and tracking in an urban environment using a multilayer laser scanner," *IEEE Trans. on Intelligent Transportation Systems*, vol. 11, pp. 579–588, 2010.

[7] A. Broggi, P. Cerri, S. Ghidoni, P. Grisleri, and H. Jung, "A new approach to urban pedestrian detection for automatic braking," *Journal of Intelligent Vehicles Systems*, vol. 10, no. 4, pp. 594–605, 2009.

[8] R. Labayrade, C. Royere, D. Gruyer, and D. Aubert, "Cooperative fusion for multi-obstacles detection with use of stereovision and laser scanner," *Autonomous Robots*, vol. 19, pp. 117–140, 2005.

[9] H. Weigel, P. Lindner, and G. Wanielik, "Vehicle tracking with lane assignment by camera and lidar sensor fusion," in *IEEE Intelligent Vehicles Symposium*, 2009, pp. 513–520.

[10] R. Altendorfer, "Observable dynamics and coordinate systems for automotive target tracking," in *IEEE Intelligent Vehicles Symposium*, 2009.

[11] M. Buhren and B. Yang, "A global motion model for target tracking in automotive applications," in *IEEE Conference on Acoustics, Speech, and Signal Processing*, vol. 2, Honolulu, Hawaii, USA, April 2007, pp. 313–316.

[12] M. Meilland, A. Comport, and P. Rives, "A spherical robot-centered representation for urban navigation," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2010, pp. 5196–5201.

[13] D. Scaramuzza, F. Fraundorfer, and R. Siegwart, "Real-time monocular visual odometry for on-road vehicles with 1-point ransac," *IEEE Int. Conf. on Robotics and Automation ICRA*, vol. 1, p. 1, 2009.

[14] B. Kitt, A. Geiger, and H. Lategahn, "Visual odometry based on stereo image sequences with ransac-based outlier rejection scheme," in *IEEE Intelligent Vehicles Symposium*, 2010, pp. 486–492.

[15] B. Kitt, F. Moosmann, and C. Stiller, "Moving on to dynamic environments: Visual odometry using feature classification," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2010, pp. 5551–5556.

[16] S. Obdrzalek and J. Matas, "A voting strategy for visual ego-motion from stereo," in *IEEE Intelligent Vehicles Symposium*, 2010, pp. 382–387.

[17] D. Pojar, P. Jeong, and S. Nedevschi, "Improving localization accuracy based on lightweight visual odometry," in *International IEEE Conference on Intelligent Transportation Systems*, 2010, pp. 641–646.

[18] S. A. Rodriguez, V. Fremont, and P. Bonnifait, "An experiment of a 3d real-time robust visual odometry for intelligent vehicles," in *IEEE Int. Conf. on Intelligent Transportation Systems*, vol. 1, Saint Louis, USA, 2009, pp. 226 – 231.

[19] A. Comport, E. Malis, and P. Rives, "Accurate quadrifocal tracking for robust 3d visual odometry," *IEEE Int. Conf. on Robotics and Automation*, pp. 40–45, April 2007.

[20] A. Petrovskaya and S. Thrun, "Model based vehicle tracking in urban environments," *IEEE Int. Conf. on Robotics and Automation, Workshop on Safe Navigation*, vol. 1, pp. 1–8, 2009.

[21] F. Nashashibi, A. Khammari, and C. Laurgeau, "Vehicle recognition and tracking using a generic multisensor and multialgorithm fusion approach," *International Journal of Vehicle Autonomous Systems*, vol. 6, pp. 134–154, 2008.

[22] Y. B. Shalom and W. D. Blair, *Multitarget/Multisensor Tracking: Applications and Advances*. Artech House Publishers, 2000.

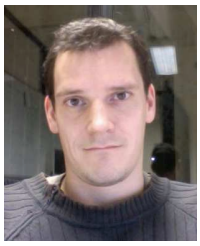
[23] M. E. Liggins, D. L. Hall, and J. Llinas, *Handbook of Multi-Sensor Data Fusion*, M. E. Liggins, D. L. Hall, and J. Llinas, Eds. CRC Press, 2008.

[24] S. A. Rodriguez, V. Fremont, and P. Bonnifait, "Extrinsic calibration between a multi-layer lidar and a camera," in *IEEE Int. Conf. on Multisensor Fusion and Integration for Intelligent Systems*, vol. 1, Seoul, Korea, 2008, pp. 214–219.

[25] J.-Y. Bouguet, "Pyramidal implementation of the lucas kanade feature tracker description of the algorithm," Intel Corporation Microprocessor Research Labs, Tech. Rep., 2002.

[26] A. Comport, E. Malis, and P. Rives, "Real-time quadrifocal visual odometry," *International Journal of Robotics Research*, vol. 29, pp. 245–266, 2010.
[27] C. V. Stewart, "Robust parameter estimation in computer vision," *Society for Industrial and Applied Mathematics*, vol. 41, no. 3, pp. 513–537, 1999.
[28] S. S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*, S. S. Blackman and R. Popoli, Eds. Artech House, Incorporated, 1999.
[29] F. Fayad and V. Cherfaoui, "Tracking objects using a laser scanner in driving situation based on modeling target shape," *IEEE Intelligent Vehicles Symposium*, vol. 1, pp. 44–49, 2007.
[30] ——, "Object-level fusion and confidence management in a multi-sensor pedestrian tracking system," *IEEE Int. Conf. on Multisensor Fusion and Integration for Intelligent Vehicles*, vol. 1, pp. 58–63, 2008.
[31] C. Simms and D. Wood, *Pedestrian and Cyclist Impact*. Springer, 2009, vol. 166, no. 1, ch. Pedestrian and Cyclist Injuries, pp. 5–30.

Sergio A. Rodríguez F. was born in Bucaramanga, Colombia, in 1982. He received his Engineer's degree in Mechatronics *cum laude* from the Universidad Autonoma de Bucaramanga, Colombia, in 2005 and his M.S. degree in Control of Systems from the University of Technology of Compiegne, France, in 2007. He is currently a Ph.D. student at the HEUDIASYC CNRS Laboratory, France.



Vincent Fremont received his M.S. degree in Automatic Control and Computer Science from the Ecole Centrale de Nantes, France, in 2000 and his Ph.D. in Automatic Control and Computer Science from the Ecole Centrale de Nantes, France, in 2003. He is an associate professor in the Department of Computer Engineering at the UTC. His research interests in the HEUDIASYC CNRS Laboratory are computer vision, camera-based calibration, 3D reconstruction and motion analysis for dynamic scenes.

Philippe Bonnifait graduated from the Ecole Superieure d'Electronique de l'Ouest, France, in 1992 and gained his Ph.D. in Automatic Control and Computer Science from the Ecole Centrale de Nantes, France, in 1997. In December 2005 he obtained the Habilitation à Diriger des Recherches from the University of Technology of Compiègne (UTC). He joined the HEUDIASYC CNRS Laboratory, France, in September 1998. Since Sept. 2007 he has been professor and head of a research group in Robotics, Automation and Embedded Systems. His current research interests are in Intelligent Vehicles and Advanced Driving Assistance Systems, with particular emphasis on dynamic ego-localisation based on multisensor-fusion and tracking.

Véronique Cherfaoui received her M.S. degree in computer science from Lille University, France, in 1988 and her Ph.D. degree in control of systems from the University of Technology of Compiegne, France in 1992. She defended an "*Habilitation à Diriger des Recherches*" in 2009. She is now an associate professor in the Computer Engineering Department at the University of Technology of Compiegne. Her research interests in the HEUDIASYC CNRS Laboratory are data fusion algorithms in distributed architecture, data association and real-time perception systems for intelligent vehicles.