2018 IEEE Intelligent Vehicles Symposium (IV)
Changshu, Suzhou, China, June 26-30, 2018

Semi-Automatic High-Accuracy Labelling Tool for Multi-Modal Long-Range Sensor Dataset

R. Izquierdo, I. Parra, C. Salinas, D. Fernández-Llorca and M. A. Sotelo

***Abstract*— Many research works have contributed to achieve SAE levels 3 and 4 in some pre-defined areas under certain restrictions. A deeper scene understanding and precise predictions of drivers intentions are needed to continue improving autonomous driving capabilities to reach higher SAE levels. Deployment of accurate and detailed datasets could be considered as one of the most pressing needs to enhance autonomous driving capabilities. This work presents a novel data acquisition methodology for on-road vehicle trajectory collection. The proposed sensor setup improves the range and detection accuracy by using a high accuracy laser scanner covering 360° and two high-speed and high-resolution cameras. The sensor fusion increases the labelling resolution and extends the detection range sporting the best of each sensor. A Median Flow tracking algorithm and a Convolutional Neural Network enable a semi-automatic labelling process, which reduces the effort to create detailed annotated datasets. High accurate trajectories are reconstructed with few manual annotations up to 60 m with a mean error below 2 cm. This methodology has been developed with a view to creating a dataset which enables the development of advanced vehicle trajectory prediction systems, and thus to contribute to human-like automated driving.**

I. Introduction and Related Works

Current advances in autonomous vehicles and active safety systems have demonstrated autonomy and safety in a wide set of driving scenarios. SAE Levels 3 and 4 have been achieved in some pre-defined areas under certain restrictions. In order to improve the level of safety and autonomy, self-driving cars need to be endowed with the capacity of anticipating potential hazards, which involves a deeper understanding of the complex driving behaviors corresponding to other human-driven cars, including inter-vehicle interactions. The need of data from naturalistic driving scenarios can be considered as one of the most important requirements for revealing, modeling and understanding driver behaviors [1] as well as for accelerating the evaluation of automated vehicles [2]. A considerable effort has been made during the last decade to collect data from equipped vehicles driven under naturalistic conditions, covering different driving tasks, such as car following, lane change, lane departure, cut-in maneuvers, etc., and using different in-vehicle sensors, such as cameras, 2D laser scanners, radars, CAN-Bus signals, GPS devices, etc. [1].

Among the different variables that can be collected, the vehicle trajectories are the most relevant features when modeling complex interactions of vehicles that share the same road section [3]. Trajectories can be collected from high-resolution cameras mounted on the infrastructure [4], [5], or on helicopters [4]. However, due to the intrinsic limitations of the cameras to obtain accurate distance measurements at far distances [6], these approaches are somehow limited to frameworks where accuracy is not critical (e.g., microscopic traffic flow models). On-road vehicle collection is a more suitable approach to obtain accurate trajectories of adjacent vehicles. For that purpose, radar sensors are by far the most common choice for data acquisition [7], [8], [9], [10], [11], [12]. Although radar sensors are very robust at detecting vehicles and measuring their distance reliably, they suffer from a poor lateral position accuracy [13]. Although this handicap can be partially mitigated by using a strong temporal filtering, it involves an increased latency which affects the reported lateral position accuracy. Thus, in most cases they are used to get the relative distance and speed of vehicles located in the same lane for car following tasks [7], [8], [9], [10]. When used for lane change studies, where lateral position accuracy is critical [14], they are combined with other sensors such as cameras [11] or 2D laser scanners [12]. The use of 2D laser scanners for on-road trajectory vehicle collection [12], [3], is the most suitable methodology to obtain accurate measurements of both longitudinal and lateral positions of surrounding vehicles. However, their use involves other limitations. For example, in order to provide

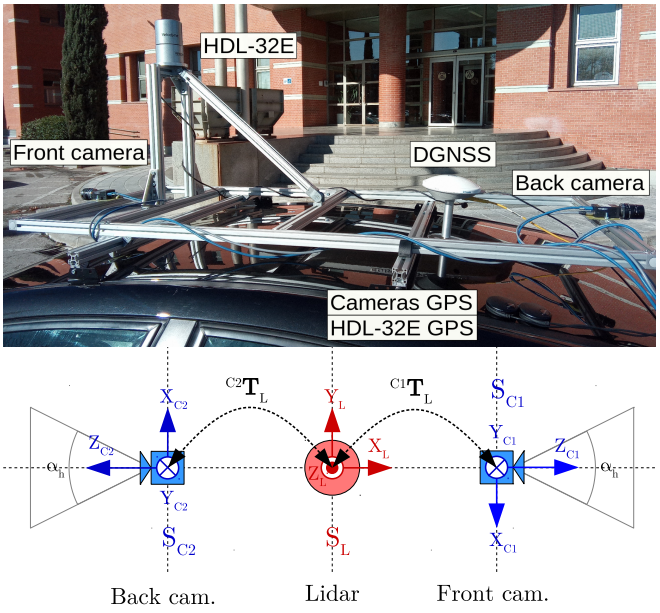

Fig. 1. Mobile platform and sensors setup top view.

R. Izquierdo, I. Parra, C. Salinas, D. Fernández-Llorca and M. A. Sotelo are with the Computer Engineering Department, University of Alcalá, Alcalá de Henares, Spain ruben.izquierdo@uah.es

360° coverage, several units have to be installed on-board the vehicle, which increases the costs and the system complexity. Thus, in [12] several side-facing radars and front- and rear-facing 2D lidars are needed, and in [3] four 2D lidars have to be used. Other limitations arises from the limited angular resolution, which compromises the effective range (∼40-50m [12], [3]), and the strong sensitivity of the horizontal detection plane to pitch variations, which compromises both the effective range and the detection robustness.

In this paper, we propose a novel data acquisition methodology for on-road vehicle trajectory collection that overcomes the limitations of current approaches in terms of complexity, accuracy and range. A new sensor setup is proposed by using a 3D lidar (Velodyne HDL-32E) which directly provides 360° coverage, in combination with two high-resolution cameras (front and rear). Once the extrinsic relationship between the cameras and the 3D lidar is calibrated, the combination of both sensors enhances the final detection range. On the one hand, low angular resolution of the 3D lidar is compensated by using the cameras, allowing an extended detection range and a more dense representation of the vehicles shape. On the other hand, depth estimation limitation of the cameras is avoided by using the highly accurate 3D lidar measurements. A novel semi-automatic labeling tool is provided to collect 360° vehicle trajectories up to distances of 60m with an angular resolution of 0.027°, which clearly outperform the current methodologies for on-road vehicle trajectory collection.

The rest of the paper is organized as follows: Section II describes the hardware used and analyzes the capabilities of the sensors. The sensors calibration process is exposed in Section III. Section IV presents the semi-automatic labelling process. Some experimental results are presented and discussed in Section V, and finally, conclusions and future works are addressed in Section VI.

II. Platform and Hardware Description

The mobile platform is a Citroën C4 [15] equipped with a rotative lidar scanner and two high-speed color cameras. The lidar is a Velodyne HDL-32E located 0.55 meters over the roof, this place is a strategic position which is where the lidar's layers are most useful. The HDL-32E data rate is approximately 32 vertical scans plus 2011 horizontal scans at 10 Hz. The used cameras are a Grasshopper3 with 12,5 mm fixed focal length lens, they are located over the roof pointing to the front and the back of the vehicle, in order to cover the maximum drivable area. The cameras work up to 1920x1200@163Hz, for this experiment the frame rate has been set to 100 Hz. Fig. 1 shows the sensors setup over the mobile platform and a schematic top view of the sensors reference systems.

A. Cameras and Lidar Range and Resolution

The goal of this work is to provide a tool able to generate a dataset for vehicle's trajectory prediction in highway scenarios. In these scenarios, long-range detection sensors are needed. A combination of different types of sensors helps to improve the detection range. In this work, a lidar and a couple of cameras are used together exploiting the best of each sensor.

One of the most relevant characteristics of the camera is the horizontal Angle Of View (AOV), which represents the area covered by the camera in the horizontal axis. This value is computed as it is shown in eq. 1, and for our camera-lens setup $\alpha_h = 48.12°$.

$$\alpha_h = 2\tan^{-1}\left(\frac{w \cdot d_x}{2f}\right) \tag{1}$$

where w is the horizontal sensor resolution, d_x is the pixel width and f is the focal distance, in this case 1920 px, 5,86 μm/px and 12.5 mm respectively.

Other important characteristic of the camera-lens setup is the horizontal Angular Resolution (AR), which basicly determines the lateral positioning resolution. Eq 2 shows the dependency of the AR with the pixel width and the focal lenght, for our camera-lens setup $\Delta\alpha_h = 0.027°$.

$$\Delta\alpha_h = \tan^{-1}\left(\frac{d_x}{f}\right) \tag{2}$$

The HDL-32E is a rotative laser-scan turning approximately at 10 Hz. Its horizontal AOV is 360° and 2011 horizontal scans are performed in each turn. The horizontal AR of the lidar can be computed dividing the horizontal AOV by the number of horizontal scans. The result is $\Delta\alpha_h \simeq 0.18°$, 6.5 times larger than in the case of the camera. This means that the camera is 6.5 times more accurate and dense than the lidar for the lateral labelling task.

Given $\Delta\alpha_h$ the lateral positioning resolution can be defined as a function of the target distance. Table I shows the lateral positioning resolution up to 100 m for the camera and the lidar.

Target Distance	**10**	**25**	**50**	**75**	**100**	[m]
Camera Resolution	2.2	5.5	10.9	16.4	21.9	[mm]
HDL-32E Resolution	15.6	39.0	78.1	117.1	156.2	[mm]

TABLE I
Lateral Positioning Resolution

The main problem using a monocular camera setup is the fact that it is not possible to determine the position of an object without assumptions. In other words, the impossibility to measure the pixel's depth, unlike stereo camera systems. A point (u, v) in the image plane reference system $\mathrm{S_I}$ defines a 3D line in the camera reference system $\mathrm{S_C}$ according to the camera pin-hole model. This means that multiple solutions are possible, but knowing one of the coordinates of the point x_C, y_C, or z_C, the others are fixed. Eq. 3 shows the pin-hole camera model, where $\mathbf{K}$ is the intrinsic camera matrix.

$$\begin{bmatrix} uw \\ vw \\ w \end{bmatrix}_\mathrm{I} = \mathbf{K} \begin{bmatrix} x \\ y \\ z \end{bmatrix}_\mathrm{C} \tag{3}$$

The HDL-32E is a high precision laser range sensor with ±2 cm accuracy which is one of the best options to solve

the indeterminacy of the pixel 3D location. Assuming the objects which will be tracked are cars, and these are mostly parallel in the front and the back to the cameras (constant z_{C} values) the lidar measures can be used to set this coordinate and solve the points indetermination problem.

HDL-32E detection range is up to 100 m, due to the lidar layers distribution, the maximum detection range over the ground plane is reduced up to 72 m without pitch variations. Fig. 2 shows the relevant lidar layer distribution regarding this limitation. The first tilted down layer has an elevation of -1.33°. This layer intersects with the ground plane approximately at 85 m, however, the vehicles are not in contact with the road on front or back bumper. Assuming a maximum clearance of 0.3 m w.r.t the ground plane the maximum detection distance decrease as far as 72 m.

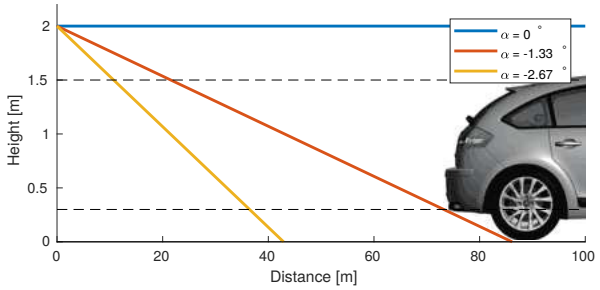

Fig. 2. Lidar detection range without pitch variations.

B. Sensors Synchronization

Multisensor acquisition architectures need a time reference framework, especially with high frame rates. Sometimes the sensors can be synchronized using hardware signals, but others, cannot. The HDL-32E spins freely and there is no control over the lidar scans. However, it is self-synchronized with the GPS time reference system. Every group of 12 horizontal scans is referenced to the GPS time. On the other hand, the cameras have an external trigger signal which is managed by a specific synchronization hardware described in [6]. This hardware is synchronized with the GPS time by a Pulse Per Second (PPS) signal and the recording computer running a Network Time Protocol (NTP) server.

III. Camera-Lidar Calibration

The camera-lidar calibration process estimates a homogeneous transformation matrix ${}^{\mathrm{C}}\mathbf{T}_{\mathrm{L}}$ that allows transforming points from the lidar reference system $\mathrm{S_L}$ to $\mathrm{S_C}$, and v/v.

In order to avoid ambiguities regarding the orientation of the calibration pattern planes, its equation has been defined in eq. 4. Subindex S represents the sensor reference system, C or L for the camera and lidar respectively.

$$\Pi_{\mathrm{S}} : ax + by + cz = d \quad d \leq 0, \quad ||(a,b,c)|| = 1 \qquad (4)$$

The camera-lidar calibration process has three steps. The first one calibrates the camera and generates Π_{C}, the second one finds Π_{L} and finally, the last step finds the best extrinsic calibration matrix ${}^{\mathrm{C}}\mathbf{T}_{\mathrm{L}}$ in a Singular Value Decomposition (SVD) fashion and a posterior non-linear optimization algorithm.

A. Camera Calibration

The camera calibration process consists of the estimation of the intrinsic matrix $\mathbf{K}$ and the lens distortion coefficients. For this purpose the Matlab® Computer Vision System Toolbox™ has been used. Knowing the parameters that model the sensor the images are undistorted and the calibration pattern equation Π_{C} is directly computed for each calibration image.

B. Lidar Calibration

The lidar calibration process, unlike the camera calibration process, needs manual inputs from the user. As an advantage, there are no restrictions to make this method works, like i.e. background constraints. It is necessary to manually mark the corners of the calibration pattern for each image-cloud pair to generate Π_{L}. This process consists of three stages. In the first one, the corners of the calibration pattern p_m are manually selected and used to compute the centroid of the calibration pattern c_m and a preliminary plane Π_m by means of Least Squares (LS). In the second step, a basic geometric segmentation is performed over the point cloud in order to isolate the calibration pattern set of points P based on condition expressed in eq. 5 where $d_{CP/2}$ is the semi-diagonal length of the calibration pattern and d_{Π} is a distance threshold to the plane.

$$p \in P \text{ if } \left(d(c_m, p) \leq d_{CP/2} \quad \& \quad d(\Pi_m, p) \leq d_{\Pi}\right) \qquad (5)$$

Finally, a closed-form robust method [16] has been used to fit the plane Π_{L} using the segmented set of points P. Firstly, a tentative plane is computed in an LS fashion using P. In every iteration $n \leq N$, the set of points P is sorted based on the distance to the plane Π_{L}, then, the η percentage of points with the longest distances are removed, and finally, a new plane is estimated with the remaining points (see Algorithm 1). Fig. 3 shows the input set of points P in blue, the final set of points used to fit the plane in green and the fitted plane in black.

$\Pi_{\mathrm{L}} = \mathrm{LeastSquares}(P)$
for $n \leq N$ **do**
  $\mathrm{sort}(P,$ based on $d\left(p_i \in P, \Pi_{\mathrm{L}}\right))$
  $P = \left\{p_0, \cdots, p_{(1-\eta)I}\right\}$
  $\Pi_{\mathrm{L}} = \mathrm{LeastSquares}(P)$
end

Algorithm 1: Plane Estimation

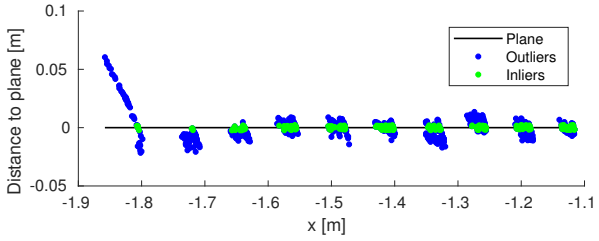

Fig. 3. Calibration pattern plane estimation. Π_{L} transformed to $z = 0$.

C. Extrinsic Calibration

The extrinsic calibration process finds a homogeneous transformation matrix ${}^{\mathrm{C}}\mathbf{T}_{\mathrm{L}}$ which transform points p_{L} from $\mathrm{S_L}$ to points p_{C} in $\mathrm{S_C}$.

$$ {}^{\mathrm{C}}\mathbf{T}_{\mathrm{L}} = \begin{bmatrix} \mathbf{R}_{3\mathrm{x}3} & \mathbf{t}_{3\mathrm{x}1} \\ \mathbf{0}_{1\mathrm{x}3} & 1 \end{bmatrix} \tag{6} $$

The conventional extrinsic calibration process uses a set of point pairs as input to estimate the homogeneous transformation matrix. However, the corners of the calibration pattern cannot be detected by the lidar and consequently, the point pairs cannot be established. An alternative calibration process is performed aligning the calibration plane pairs.

The n plane equations are used to compute matrix $\mathbf{N}$ and vector $\mathbf{d}$ which represent the normal plane vectors (a, b, c), and the distances to the origin of the reference system d.

$$ \mathbf{N} = \begin{bmatrix} a_1 & b_1 & c_1 \\ \vdots & \vdots & \vdots \\ a_n & b_n & c_n \end{bmatrix}, \quad \mathbf{d} = \begin{bmatrix} d_1 \\ \vdots \\ d_n \end{bmatrix} \tag{7} $$

With this notation the sets of plane equations Π_{C} and Π_{L} can be rewriten as it is shown in eq. 8 and 9 respectively.

$$ \mathbf{N}_{\mathrm{C}} \cdot p_{\mathrm{C}} = \mathbf{d}_{\mathrm{C}} \tag{8} $$

$$ \mathbf{N}_{\mathrm{L}} \cdot p_{\mathrm{L}} = \mathbf{d}_{\mathrm{L}} \tag{9} $$

The aligning process firstly estimates the traslation vector $\mathbf{t}$. Eq. 6 can be rewriten as $p_{\mathrm{C}} = \mathbf{R} \cdot p_{\mathrm{L}} + \mathbf{t}$. The only one point which is not affected by the rotation is $0_{\mathrm{L}} = (0, 0, 0)$, appliying the traslation vector to 0_{L} we have $p_{\mathrm{C}} = \mathbf{t}$. Replacing p_{C} by $\mathbf{t}$ and p_{L} by 0_{L}, and substracting eq. 9 to eq. 8 the result is:

$$ \mathbf{N}_{\mathrm{C}}\mathbf{t} = \mathbf{d}_{\mathrm{C}} - \mathbf{d}_{\mathrm{L}} \tag{10} $$

Eq. 10 can be solved by Least Squares and the translation vector $\mathbf{t}$ is achieved.

The second step computes the rotation matrix $\mathbf{R}$ in an SVD fashion. Basically, the rotation matrix must align the normal vectors of the plane pairs as it is shown in eq. 11.

$$ \begin{aligned} \mathbf{N}_{\mathrm{C}}^T &= \mathbf{R}\mathbf{N}_{\mathrm{L}}^T \\ \mathbf{U}\mathbf{S}\mathbf{V}^T &= \mathbf{N}_{\mathrm{L}}^T\mathbf{N}_{\mathrm{C}} \\ \mathbf{V}\mathbf{U}^T &= \mathbf{R} \end{aligned} \tag{11} $$

Finally, a non-linear optimization algorithm finds the minimum of an unconstrained multivariable function using derivative-free method [17]. $\mathbf{R}$ is transformed into an unconstrained vector using a quaternion transformation. The translation vector $\mathbf{t}$ is concatenated to the quaternion. The cost function for the optimization routine is defined as it is shown in eq. 12 where i represents the image-cloud pair and ${}^{\mathrm{C}}\mathbf{T}_{\mathrm{L}}^*$ is the simplified trasformation matrix

$$ f(\mathbf{R}, \mathbf{t}) = \sum_{i=0}^{n} ||d\left(\Pi_{\mathrm{C},i}, {}^{\mathrm{C}}\,\mathbf{T}_{\mathrm{L}}^* \cdot p_{m,i}\right)|| \tag{12} $$

When $\mathbf{K}$ and ${}^{\mathrm{C}}\mathbf{T}_{\mathrm{L}}$ are computed, p_{L} can be transformed to $\mathrm{S_C}$ and then into $\mathrm{S_I}$ according with eq. 13.

$$ \begin{bmatrix} uw \\ vw \\ w \end{bmatrix}_{\mathrm{I}} = \mathbf{K}{}^{\mathrm{C}}\mathbf{T}_{\mathrm{L}}^* \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{\mathrm{L}} \tag{13} $$

Fig. 4 shows the projection of the color image over the lidar scans.

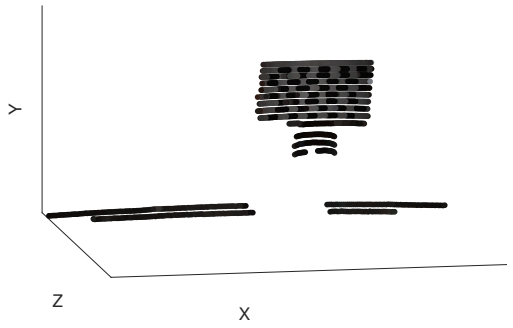

Fig. 4. Image color reprojection over the lidar scans.

IV. Labelling and Tracking

As we expose in section II a point in the image plane defines a 3D line in the camera reference system. Labelling a target point in the image sets the target direction. Assuming that the vehicles are on the ground and the ground is mostly flat on highway scenarios this direction can be interpreted as the angular coordinate in a polar reference system. Finally, the depth, or radial coordinate, is estimated using the lidar detections. The point used as the reference for the vehicle location is any point horizontally centered located on the front or rear bumper. The best key points for a correct vehicle location are the logo, the license plate or any other feature symmetrically placed on the vehicle.

A. Image Tracking

A state of the art Median-Flow tracker algorithm [18] has been used in order to semi-automate the labelling process. Firstly, the tracker is set up with a Region Of Interest (ROI) containing the desired key points and consecutively updated. This labelling tool implements a multi-object-multi-camera tracking system with key points reset option. The tracked key points can be changed when the tracker lost them or when the user considers that the tracker accuracy is not enough.

Fig. 5 shows the tracking of a front and back vehicle key point along 3 seconds which represents 300 images and 30 image-cloud pairs. The green box represents the bounding box of the tracked key point and the red mark is the manual annotated key point.

B. Depth Estimation

Once the vehicles or a part of them are labelled it is necessary to assign a depth value or z coordinate to this region. A semantic segmentation FCN-8s Convolutional Neural Network (CNN) [19] has been used in order to segment the

Fig. 5. Vehicle key point tracking. From top to bottom, tracker initialization, and key points after 1, 2 and 3 seconds of tracking using 100 FPS. Left column was tracked in reverse recording order and right column in the forward. 1, 2, 4 and 8 zoom factor have been applied in each row.

area of the vehicles in the image. For each annotation the vehicle container area is searched into the segmented image, setting the shape of the annotated vehicle in the image. Fig. 6 shows an example of the vehicle semantic segmentation. Using eq. 13 the lidar points are reprojected over the image and the longitudinal distance to the vehicle is estimated using the closest lidar detection over the vehicle shape in the image.

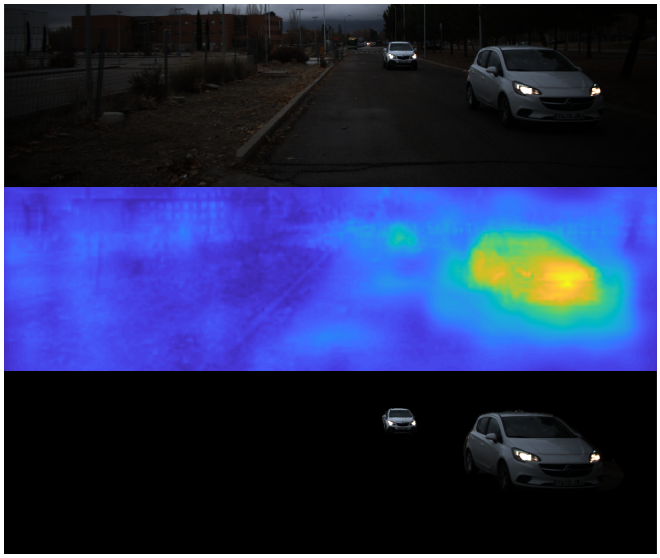

Fig. 6. Vehicle semantic segmentation. From top to bottom, original image, vehicle class heat map and segmented vehicles. Warm colors represent higher probabilities.

V. Results

In this section, the final results are shown and commented. Figure 7 shows the tracking errors on the image plane. In the left sub-trajectory, there is a tracking reset event marked with a red circle due to the tracking drift. In the right sub-trajectory, the error rises since the point marked with a red square due to a partial occlusion which displaces the tracker. The tracking Mean Root Square Error (MRSE) and the Mean Absolute Error (MAE) for this trajectory are shown in table II.

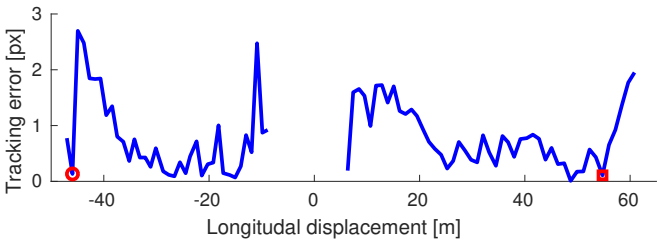

Fig. 7. Tracking errors on image plane.

Fig. 8 shows the trajectory reconstruction. The green line represents the best possible result which will be considered as ground truth for error metric computation. It is achieved using manual annotations in both the image and the point cloud. The red line represents manual annotations using only the point cloud. The blue line is the semi-automatic trajectory reconstruction. As can be seen in fig. 8 and table II the presented methodology overcomes the results generated using only the lidar. This is a proof of the sensor fusion capability to increase the detection accuracy due to the higher angular resolution in the camera sensor w.r.t the lidar.

The trajectory generated by the semi-automatic method is quite similar to the manually annotated one. The error metrics are shown in table II. There are small errors at the beginning of the left sub-trajectory because of the tracking error mentioned previously. The lateral error is reduced to zero when the tracking is reset. This instant is represented by a black circle in fig. 8. On the other hand, there is a small difference at the end of the right sub-trajectory. The trajectory starts to differ in the point represented by the black square due to the visual occlusion of the tracked key point.

Method	MRSE	MAE	σ	
Point cloud	18.67	15.35	14.84	[cm]
Semi-automatic	2.42	1.65	2.32	[cm]
Tracking	1.00	0.80	0.61	[px]

TABLE II
Lateral Position Error Metrics

VI. Conclusions and Future Works

The presented semi-automatic labelling method improves the detection range and the angular resolution with 360° coverage exploiting advantages of a multi-modal sensors setup. A camera-lidar calibration method enables the sensor

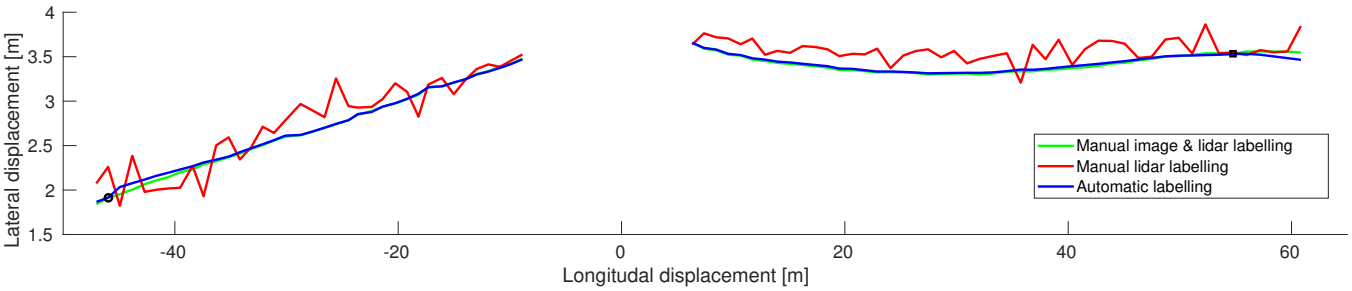

Fig. 8. Trajectory reconstruction.

data fusion. The current detection range is up to 60 meters due to the angular lidar configuration and the angular labelling resolution is 0.027° with a horizontal AOV of 48° in the front and the back. This methodology enhances range and precision of datasets and enables the basis for future studies on driver behavior predictions. This method is vulnerable to visual occlusions from the sensors point of view. The use of radar technology could solve this issue and extend the detection range. Another contribution is the semi-automatic supervised process for labelling tasks. A state of the art tracking algorithm and a semantic segmentation CNN automate the labelling process.

As future works, an automatic calibration pattern extraction from the point clouds would become the camera-lidar calibration in a fully automatic method. The camera-lidar calibration could be improved inferring the lidar pattern points using the scans intensity and then performing point-pairs-based calibration. The adjoining lanes are not covered by the cameras in the proximity of the vehicle, a vehicle lidar detection in this area is needed to complete the transition from one camera to the other.

VII. ACKNOWLEDGEMENTS

This work was funded by Research Grants SEGVAUTO S2013/MIT-2713 (CAM), DPI2017-90035-R (Spanish Min. of Economy), BRAVE Project, H2020, Contract #723021 and FPU14/02694 (Spanish Min. of Education) via a predoctoral grant to the first author. This project has received funding from the Electronic Component Systems for European Leadership Joint Undertaking under grant agreement No 737469 (AutoDrive Project). This Joint Undertaking receives support from the European Unions Horizon 2020 research and innovation programme and Germany, Austria, Spain, Italy, Latvia, Belgium, Netherlands, Sweden, Finland, Lithuania, Czech Republic, Romania, Norway.

REFERENCES

[1] W. Wang, C. Liu, and D. Zhao, "How much data are enough? a statistical approach with case study on longitudinal driving behavior," *IEEE Transactions on Intelligent Vehicles*, vol. 2, no. 2, pp. 85–98, 2017.

[2] D. Zhao, H. Lam, H. Peng, S. Bao, D. J. LeBlanc, K. Nobukawa, and C. S. Pan, "Accelerated evaluation of automated vehicles safety in lane-change scenarios based on importance sampling techniques," *IEEE transactions on intelligent transportation systems*, vol. 18, no. 3, pp. 595–607, 2017.

[3] H. Zhao, C. Wang, Y. Lin, F. Guillemard, S. Geronimi, and F. Aioun, "On-road vehicle trajectory collection and scene-based lane change analysis: Part i," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 1, pp. 192–205, 2017.

[4] V. G. Kovvali, V. Alexiadis, P. Zhang, *et al.*, "Video-based vehicle trajectory data collection," Tech. Rep., 2007.

[5] S. Álvarez, D. F. Llorca, and M. Sotelo, "Hierarchical camera auto-calibration for traffic surveillance systems," *Expert Systems with Applications*, vol. 41, no. 4, pp. 1532–1542, 2014.

[6] D. F. Llorca, C. Salinas, M. Jimenez, I. Parra, A. Morcillo, R. Izquierdo, J. Lorenzo, and M. Sotelo, "Two-camera based accurate vehicle speed measurement using average speed at a fixed point," in *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*. IEEE, 2016, pp. 2533–2538.

[7] S. Lefèvre, A. Carvalho, and F. Borrelli, "A learning-based framework for velocity control in autonomous driving," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 1, pp. 32–42, 2016.

[8] B. Higgs and M. Abbas, "Segmentation and clustering of car-following behavior: Recognition of driving patterns," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 1, pp. 81–90, 2015.

[9] G. Qi, Y. Du, J. Wu, N. Hounsell, and Y. Jia, "What is the appropriate temporal distance range for driving style analysis?" *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 5, pp. 1393–1403, 2016.

[10] L. Pariota, G. N. Bifulco, and M. Brackstone, "A linear dynamic model for driving behavior in car following," *Transportation Science*, vol. 50, no. 3, pp. 1032–1042, 2015.

[11] K. Nobukawa, S. Bao, D. J. LeBlanc, D. Zhao, H. Peng, and C. S. Pan, "Gap acceptance during lane changes by large-truck driversan image-based analysis," *IEEE transactions on intelligent transportation systems*, vol. 17, no. 3, pp. 772–781, 2016.

[12] V. A. Butakov and P. Ioannou, "Personalized driver/vehicle lane change models for adas," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 10, pp. 4422–4431, 2015.

[13] M. Nishigaki, S. Rebhan, and N. Einecke, "Vision-based lateral position improvement of radar detections," in *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*. IEEE, 2012, pp. 90–97.

[14] R. Izquierdo, I. Parra, J. Munoz-Bulnes, D. Fernández-Llorca, and M. Sotelo, "Vehicle trajectory and lane change prediction using ann and svm classifiers," in *Intelligent Transportation Systems (ITSC), 2017 20th International IEEE Conference on*. IEEE, 2017, pp. 1141–1146.

[15] I. P. Alonso, R. I. Gonzalo, J. Alonso, Á. García-Morcillo, D. Fernández-Llorca, and M. Á. Sotelo, "The experience of drivertive-driverless cooperative vehicle-team in the 2016 gcdc," *IEEE Transactions on Intelligent Transportation Systems*, 2017.

[16] M. Zuliani, "Ransac for dummies (2008)," *With examples using the RANSAC toolbox for Matlab and more*, 2009.

[17] T. F. Coleman and Y. Li, "An interior trust region approach for nonlinear minimization subject to bounds," *SIAM Journal on optimization*, vol. 6, no. 2, pp. 418–445, 1996.

[18] Z. Kalal, K. Mikolajczyk, and J. Matas, "Forward-backward error: Automatic detection of tracking failures," in *Pattern recognition (ICPR), 2010 20th international conference on*. IEEE, 2010, pp. 2756–2759.

[19] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr, "Conditional random fields as recurrent neural networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1529–1537.