Target Position and Speed Estimation Using LiDAR

Enes Dayangac(✉), Florian Baumann, Josep Aulinas, and Matthias Zobel

ADASENS Automotive GmbH, Oberhof 13, 88138 Weissensberg, Germany
{enes.dayangac,florian.baumann,josep.aulinas,matthias.zobel}@adasens.com
http://www.adasens.com

Abstract. In this paper, an efficient and reliable framework to estimate the position and speed of moving vehicles is proposed. The method fuses LiDAR data with image based object detection algorithm output. LiDAR sensors deliver 3D point clouds with a positioning accuracy of up to two centimeters. 2D object data leads to a significant reduction of the search space. Outliers removal techniques are applied to the reduced 3D point cloud for a more reliable representation of the data. Furthermore, a multi-hypothesis Kalman filter is implemented to determine the target object's speed. The accuracy of the position and velocity estimation is verified through real data and simulation. Additionally, the proposed framework is real-time capable and suitable for embedded-vision related applications.

Keywords: LiDAR · Velodyne · Kalman filter · Multi-hypotheses · 3D point cloud · Sensor fusion · Distance · Speed estimation

1 Introduction

The automotive industry integrates varieties of peripheral sensors into advanced driver assistance systems (ADAS), e.g., ultrasonic sensors, camera sensors, mid and long range radar and LiDAR. Redundancy in the data can be exploited by sensor fusion, leading to an accurate observation of the state of the vehicle and its surroundings.

LiDAR technology is advancing and provides a full line of sensors capable of delivering accurate real-time 3D data up to 2.2 million points per second. However, it is challenging to process the resulting point cloud to retrieve relevant information in real time. Also, LiDAR is incapable of sensing visual cues such as lines, color, and brightness differences e.g. for the identification of traffic lights, brake lights, turning signals or signs. Unlike LiDAR, camera systems are able to detect and recognize objects such as traffic signs, pedestrians, or vehicles.

In this paper, a framework is presented where the distance to the target object and the target speed can be reliably estimated by integrating the data collected from LiDAR and camera sensors. 2D object detection and tracking systems are deployed to detect objects in the camera data and reduce the search space in the

A. Campilho and F. Karray (Eds.): ICIAR 2016, LNCS 9730, pp. 470–477, 2016.
DOI: 10.1007/978-3-319-41501-7_53

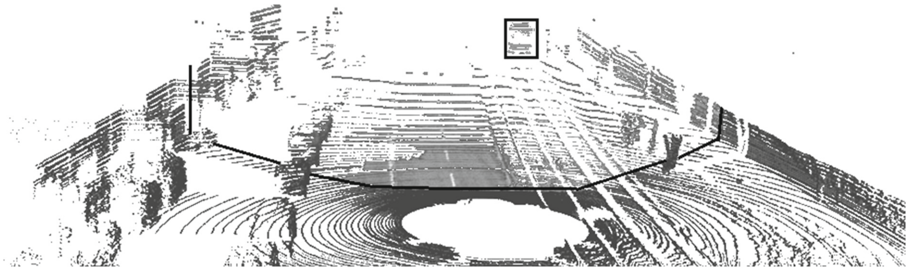

Fig. 1. A LiDAR 3D point cloud is illustrated. In the 2D image (from the KITTI data [1], 2011-09-26, drive 001 image sequence), the object is detected and overlapped with 3D data to reduce the search space. The bold line indicates the camera field of view and the box represents the vehicle in the point cloud.

3D point cloud (see Fig. 1). Along with outlier removal, filtering techniques are implemented to estimate the target speed accurately. The framework is tailored for vehicle detection. Also, it is applicable to any kind of target objects such as pedestrians, cyclists, or vehicles to enhance ADAS, e.g., forward collision warning, crossing traffic detection, or advanced emergency braking.

This paper is structured as follows. Section 2 reviews related work and Sect. 3 presents the proposed approach including fundamentals. Experimental results are shown in Sect. 4 and conclusions are drawn in Sect. 5.

2 Related Work

Distance and speed estimation with LiDAR sensors and the combination with an object detection system is a challenging research area in computer vision. Regarding the automotive industry and autonomous driving scenarios, LiDAR systems are becoming the next generation of driver assistance systems [2,3] and have high potential for achieving outstanding results.

Most of the research combines several redundant techniques to obtain a more accurate and robust system. For instance, Mählisch et al. propose a sensor fusion framework between video and LiDAR data, used to improve vehicle detection [4]. Premebida et al. propose a LiDAR and vision based approach for pedestrian and vehicle detection and tracking [5]. Ogawa et al. propose a system for pedestrian tracking and detection [6], solely using an in-vehicle LiDAR sensor. Mählisch et al. combine LiDAR, video and ESP data for vehicle tracking to enhance adaptive cruise control [7].

In comparison to these approaches, we propose a framework that combines LiDAR data with an object detection system. The contribution of this paper is to reduce the complexity in 3D data for a robust estimation of an object's distance and speed. According to the best of our knowledge, this approach has not yet been investigated.

3 Robust and Accurate Estimation

An overview of our proposed method is illustrated in Fig. 2. The general architecture can be divided into three parts: data association, outlier removal and estimation.

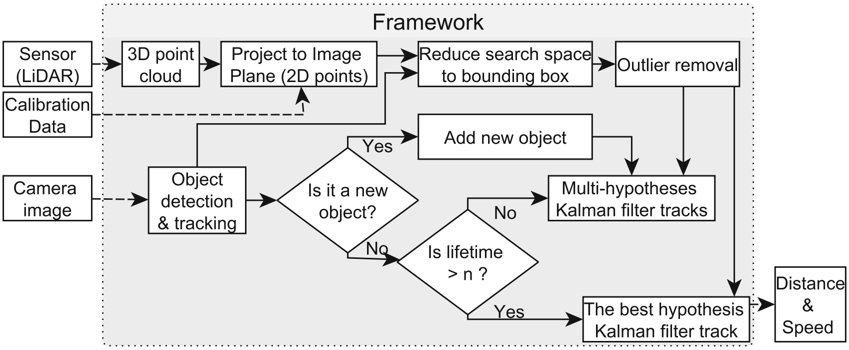

Fig. 2. Proposed framework system architecture.

3.1 Data Association

3D Mapping: 3D information is given by LiDAR, which provides a set of 3D data points. The 3D points $X_i = [x, y, z]^T$, $i \in \mathbb{N}$, $i \leq n$ where i is the point index and n the number of points in the cloud with respect to the LiDAR coordinate system. These points are transformed first to the camera coordinate system and then to the image coordinate system. In order to compute these transformations, each 3D point $X_i' = [x, y, z, 1]^T$ is represented as an homogeneous point. Let $T_{4\times4}$ be the homogeneous transformation matrix from the LiDAR to the camera coordinate system and $P_{3\times4}$ is the 10 degrees of freedom camera projection matrix. The pinhole camera model [8] is used for mapping from LiDAR to image plane in Eq. 1.

$$x_i = PTX_i' \tag{1}$$

Search Space Reduction: There are millions of unstructured points in LiDAR data. Thus, it is costly to process LiDAR points such as filtering out the points on the ground plane or segmenting objects to generate target hypotheses. For real time processing, it is required to reduce the search space. In our method, the points are filtered in the camera scene. Additionally, the output bounding box from the object detection and tracking is used to decrease the search space. Using this reduced search space, the indices of 3D points are associated to a bounding box representing an object.

Data: 3D points $X_i = [x, y, z]^T$ per bounding box
Result: mean(inliers)
$\mu = \text{mean}(X_i)$; $\sigma = \text{standardDeviation}(X_i)$
$[bin1, bin2]$= kMeans$(x, z, \|X_i\|)$
Choose the bin having larger number of points, bin = max(bin1, bin2)
while *not at end of the bin* **do**
  get next point X_j from the bin
  $k = \frac{\|X_j - \mu\|}{\sigma}$
  if $k < threshold$ **then**
    set X_j as inlier
  end
end

Algorithm 1. Outliers removal based on k-Means clustering and the ratio of absolute deviation to standard deviation.

3.2 Outlier Removal

In the 3D points belonging to an object, the outliers exist due to the following issues: The cluttered backgrounds happens at the edge of the detected bounding boxes, or the occlusion causes foreground outliers (see Fig. 3). Another issue is that target objects, such as vehicles, have glass windows where LiDAR 3D points becomes inaccurate in positioning. Finally, the detected bounding boxes can be imprecise, bigger or smaller. Therefore, a smaller Region of Interest (ROI) is considered to represent the object in detected bounding boxes. Consequently, a k-Means clustering algorithm [9] is applied and the ratio of absolute deviation to standard deviation is checked to determine outliers as shown in Algorithm 1.

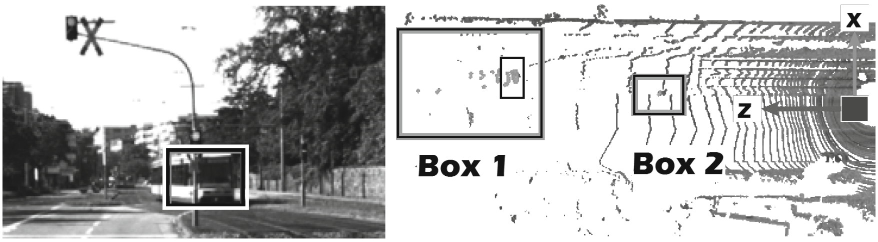

Fig. 3. Left: The box represents the detection of an object which is partially occluded by a traffic light (from KITTI data [1] 2011-09-26, drive 001 image sequence). Right: Top view LiDAR point cloud. The detected object is represented by Box 1 which has a cluttered background due to imprecise object detection. In Box 2, outliers exist due to occlusion.

3.3 Estimation

In this framework, the object detection and tracking algorithms detect and track an object over a number of frames. The algorithms locate the object and assigns an identification number to the same object through out an image sequence. The approach relies on these algorithms with the assumption of having a lower false alarm rate in urban environments. Using the identification number and detection, the corresponding 3D points are chosen. Based on the 3D points belonging to the object, the goal is to measure the object's position and speed accurately.

Kalman Filter: An extended four-state Kalman filter [10] is designed using a constant velocity model. The main requirement for the filter is to produce reliable estimations of position and speed of target vehicles. Therefore, the state vector consists of: Longitudinal distance z and lateral offset x of the target vehicle w.r.t. the ego vehicle coordinate system, relative speed $\dot{z}$ in Z-direction and $\dot{x}$ in X-direction between target and ego vehicles.

Distance observations from LiDAR are used as measurements in order to update the states. All these steps are standard and commonly used for generic object tracking in moving scenarios. The filter is required to be designed and parametrized to converge to the actual values quickly. Besides, the speed of such convergence is highly influenced by state vector initialization.

Multiple Tracks: Implementing several filters with different initial hypotheses might improve convergence time, which is critical in object detection for automotive applications. Therefore, multi-hypotheses Kalman filter tracks are initialized when a new target object is detected. That is, for each target object, multiple Kalman filters $\{K_0, K_1, K_2, ... , K_m\}$ are initialized at initial speeds v_i. The subscript m is the total number of filters and v_i ranges between values $V_{range} = [V_{min}, V_{max}]$ using a step value $\Delta v = \frac{|V_{max} - V_{min}|}{m}$.

If there is a target object having the position $p_{current}$, we can simply use the speed hypothesis and predict the next position $p_{current}$ using the formula $p_{next} = p_{current} + v_i dt$. Compared to the observation, incorrect speed hypotheses will yield bigger errors in the Kalman filter prediction.

As soon as the lifetime of an object is bigger than n, the error function $f_{error}(x, z)$ in Eq. 2 is taken into account to compute the probability to score all the filter tracks. After n frames, the filter with the highest probability (the smallest state error) is used to estimate the trajectory.

$$f_{error}(x, z) = |x_{prediction} - x_{observation}| + |z_{prediction} - z_{observation}| \tag{2}$$

A Kalman filter's complexity is $O(n^2)$ and adding additional filters for multiple tracks increases the complexity linearly. Thus, n and V_{range} can be chosen arbitrarily depending on the application.

4 Experiments

In this section, evaluation results are presented. For the evaluation, a series of experiments are conducted using both real sensor data and synthetic data in order to assess the performance. The filter state error for different hypotheses and their run-time performances are measured. Also, average position and speed errors are explored using a varying number of hypotheses and lifetime while tuning the observation noise level.

For the sensor data, the LiDAR operates at 10 Hz and generates over one million points per second. Also, a simulation script is set up to generate synthetic data with Gaussian noise. Additionally, the proposed algorithms run on a single core of an Intel i7-2630QM CPU at 2.00 GHz.

Table 1. Average run-time per target object ($V_{range} = [-30, 30]$ m/s, $\Delta v = 0.25$).

Data association	62.50 ms
Outlier removal	0.37 ms
Estimation	1.94 ms

Experimental Results: The run-time performances are inspected on real sensor data deploying 240 hypotheses per target object according to Sect. 3. The run-time results are presented in Table 1 where data association, outlier removal and estimation take 62.50 ms, 0.37 ms and 1.94 ms respectively.

The different initial hypotheses were enforced to improve the convergence time for more accurate estimation in Sect. 3.3. In Fig. 4 (left), some of the hypotheses are chosen and shown where initial speeds are $v_1 = -30$, $v_2 = -18.75$, $v_3 = -7.5$, $v_4 = 0.25$, $v_5 = 3.75$ m/s. It is clear that the filter state error f_{error} differs according to given initial speeds. In Fig. 4 (left), v_4 is a better estimation which converges faster than others, whereas v_1 has the biggest state error between observation and estimation. The number of hypotheses vs. FPS (Frames per Seconds) graph is drawn in Fig. 4 (right), where more hypotheses mean fewer FPS.

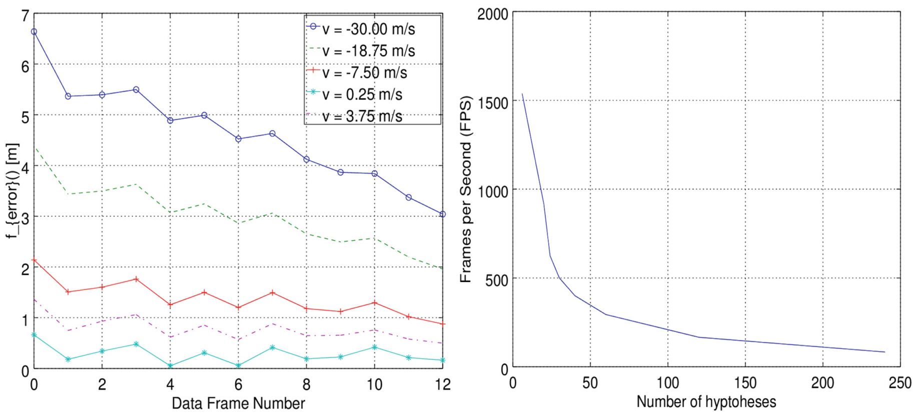

Fig. 4. Left: Different hypotheses yield varying state error f_{error} over data frame number where real relative speed is around 0 m/s. Right: FPS vs. Number of hypotheses.

Tracking an object with multiple hypotheses and the estimation with the highest probability is used for updating the target position and speed. The chosen lifetime for multiple hypotheses affects the error according to the simulation in Fig. 5. Also, larger number of hypotheses decrease the average position and speed error as shown in Fig. 6. However there is only small difference between 60, 120 and 240 hypotheses.

In the experiments using the real data, selected 3D points that belong to a target object have an average standard deviation of $\sigma_z = 0.15$ m in longitudinal distance. The average standard deviation in all directions is 0.6 m. According to this, the position estimation is in the range of $0.1 - 0.3$ m accuracy in Fig. 5.

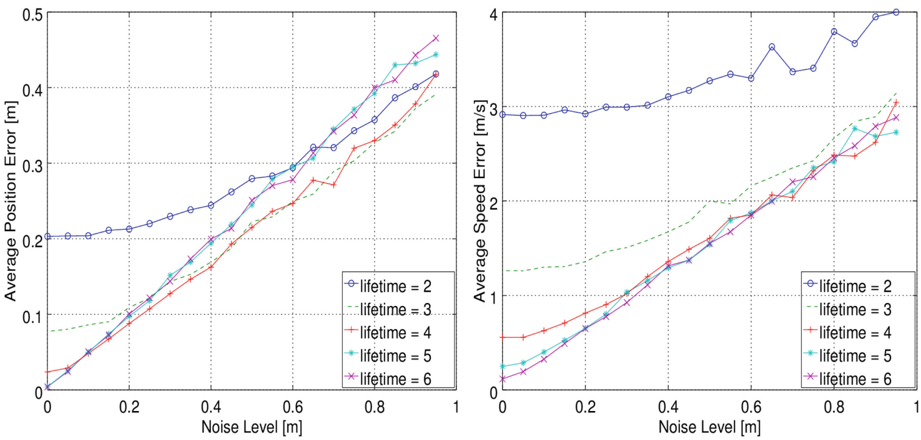

Fig. 5. Average system error using different target lifetimes and noise levels with 240 hypotheses.

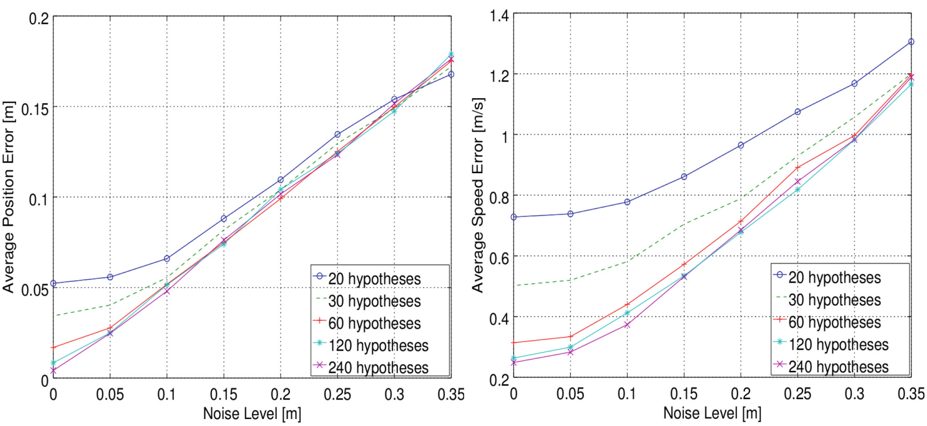

Fig. 6. Average system error using different number of hypotheses and noise levels at lifetime = 5.

5 Conclusions

A sensor fusion strategy is presented to estimate the position and the speed of target objects from LiDAR data. The 3D search space is massively reduced by assuming that 3D points that belong to a target object will be reprojected within the 2D object bounding box on the image plane. Let the noise level be 0.2 m in Fig. 6, the experimental results demonstrate that position and speed can be estimated with an accuracy of up to 0.1 m and 0.7 m/s. Furthermore, our approach produces consistent estimates after five frames. Future work should focus on optimizing the data association stage, which is currently the main bottleneck of our proposed method. The remaining stages of this approach, e.g., outlier removal, filtering, position and distance computation, are proven to be real-time suitable for embedded-vision purposes in ADAS applications.

References

1. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. Int. J. Robot. Res. (IJRR) (2013)
2. Rasshofer, R., Gresser, K.: Automotive radar and lidar systems for next generation driver assistance functions. Adv. Radio Sci. **3**(B.4), 205–209 (2005)
3. Ghring, D., Wang, M., Schnrmacher, M., Ganjineh, T.: Radar/lidar sensor fusion for car-following on highways. In: Automation, Robotics and Applications (ICARA), pp. 1838–1845 (2011)
4. Mählisch, M., Schweiger, R., Ritter, W., Dietmayer, K.: Sensorfusion using spatio-temporal aligned video and lidar for improved vehicle detection. In: IEEE Intelligent Vehicles Symposium (2006)
5. Premebida, C., Monteiro, G., Nunes, U., Peixoto, P.: A lidar and vision-based approach for pedestrian and vehicle detection and tracking. In: Intelligent Transportation Systems Conference (ITSC), pp. 1044–1049. IEEE (2007)
6. Ogawa, T., Sakai, H., Suzuki, Y., Takagi, K., Morikawa, K.: Pedestrian detection and tracking using in-vehicle lidar for automotive application. In: Intelligent Vehicles Symposium, pp. 734–739. IEEE (2011)
7. Mählisch, M., Hering, R., Ritter, W., Dietmayer, K.: Heterogeneous fusion of video, lidar and esp data for automotive acc vehicle tracking. In: Multisensor Fusion and Integration for Intelligent Systems, pp. 139–144. IEEE (2006)
8. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, New York (2003)
9. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. Wiley-Interscience, New York (2000)
10. Kalman, R.E.: A new approach to linear filtering and prediction problems. J. Basic Eng. **82**(1), 35–45 (1960)