LiDAR Based Real Time Multiple Vehicle Detection and Tracking

Zhongzhen Luo, Saeid Habibi, Martin v. Mohrenschildt

Abstract—Self-driving vehicle require a high level of situational awareness in order to maneuver safely when driving in real world condition. This paper presents a LiDAR based real time perception system that is able to process sensor raw data for multiple target detection and tracking in dynamic environment. The proposed algorithm is nonparametric and deterministic that is no assumptions and priori knowledge are needed from the input data and no initializations are required. Additionally, the proposed method is working on the three-dimensional data directly generated by LiDAR while not scarifying the rich information contained in the domain of 3D. Moreover, a fast and efficient for real time clustering algorithm is applied based on a radially bounded nearest neighbor (RBNN). Hungarian algorithm procedure and adaptive Kalman filtering are used for data association and tracking algorithm. The proposed algorithm is able to run in real time with average run time of 70ms per frame.

Keywords—LiDAR, real-time system, clustering, tracking, data association.

I. INTRODUCTION

EFFECTIVE and reliable environment perception is one of an important role in many intelligent vehicle applications such as self-driving vehicle and many other vehicle control applications. Although camera based system has outstanding features that using sophisticated techniques and low operation cost, it is less effective under complex circumstance such as bad weather conditions [1]. Radar has advantages of longer range detection and higher reliability but low angular resolution constrains its field of view [2]. Within the last few years, fully three-dimensional laser scanners have been introduced. Rather than scanning in 2D space, 3D space is scanned resulting in a cloud of 3D points. A Light Detection and Ranging (LiDAR) based on laser scanner has proven efficient due to its high accuracy in ranging, its wide-area view, and low data-processing requirements. Additionally, theirs capabilities in adverse weather conditions have also been improved [3]. However, processing the large amount of 3D data points is a great challenge that demanding efficient algorithms and fast data structures. In DARPA Urban Challenge, the laser 3D data is usually converted into lower dimension. For example, by using a $2\frac{1}{2}$ occupancy grid map to project 3D points to a 2D plane where segmentation and classification of objects are processed in terms of occupancy grid map [4], [5]. One of the advantages is that several sensors can be fused easily and the mapping strategy will become straight-forward. Segmentation of 3D point clouds is a another critical step in a number of environmental

Zhongzhen Luo is with the Department of Software Engineering, McMaster University, Hamilton, Canada (e-mail: philippeluo@gmail.com).

perception tasks. A simple solution is to define the density of the points within a cell as occupant value [4]. This easily extract features from each cell, yielding connected components not belonging to the ground surface. Base on the same compressed data in a $2\frac{1}{2}$ occupancy grid, Frank and Oliver presents a segmentation algorithm in terms of the concept of local convexity which scan 4-neighborhood surface and compared their attributes such as normal vector [6]. Object separation is then usually performed by applying clustering algorithm. Douillard and Underwood compare a set of clustering techniques designed for different 3D point clouds [7]. The best selected candidate they propose is the cluster-all with variable neighborhood method where points are partitioned by the local voxel adjacency only and the size of the local neighborhood is the only parameter. However, the computation time is still very high considering for the real time application. In [8], an algorithm is presented that efficiently segments a given 3D point cloud using a radially bounded nearest neighbor (RBNN) cluster method while maintain its ability for real time processing based on the static *kd-tree*. Data association is then performed to identify measurements from a sequence of frames belonging to the same target. In [9], multiple hypothesis tracking (MHK) is adopted to handle the challenging problem of tracking multiple vehicles in urban environment with dense traffic. But the performance in time-consuming would not be able to compute in real time. Robustness and stable tracking of vehicle is another challenge to be concerned, the popular Kalman filter(KF) was introduced in the 1960s and remains one of the most popular estimation methods since it yields a statistically optimal solution for linear systems and measurements [10].

The developed method is decomposed into three steps: Segmentation, clustering and tracking (see Fig. 1). The first step is ground segmentation which is useful for object classification and dynamic obstacle detection and tracking. As a result, an accurate segmentation in different scenes is generated. In the second clustering step, the nearest neighbor clustering algorithm is presented to distinguish the generated non-ground point set. We chose kd-tree as spatial data structure because it offers competitive look-up times for radially bounded queries. In order to reduce computation time such as the number of queries as much as possible, RBNN is used to achieved this. Given the selected clusters, the tracking method needs to retrieve real objects among those targets and sets an identification for each of them in every frame. To deal with such data association problems, the Hungarian algorithm is adopted get an optimal matching between targets and detected objects. Our tracking method is adaptive KF

based on covariance matrix estimation. The evaluation results will illustrate that the developed approach achieves a close performance to the state-of-art techniques, which will have great potentials in future autonomous navigation systems.

This paper is structured as follows: Section II describes the technology utilized in this research. Section III explains the overview of proposed approach. Section III *A* explain the spatial data structure, Section III *B* illustrates segmentation algorithm, Section III *C* describes the refinement version of radially bounded nearest neighbor algorithm (RBNN) for our real-time application. Section III *D* demonstrates data association and adaptive KF is achieved to solve linear assignment problem in different frames. Experiential result in Section IV and conclusion in Section V are provided.

II. TECHNOLOGY

As an introduction to LiDAR, a basic understanding of laser is required. The word "laser" stands for "light amplification by stimulated emission of radiation". A laser is a device which generates a stream of high energy particles (photons) within an extremely narrow range of wavelengths. A laser light source forms the basis for a LiDAR system. The wavelength chosen for most lasers is 905 nanometers, which is in the near-infrared band of the electromagnetic spectrum [11]. LiDAR uses pulsed lasers to rapidly create a three dimensional image or map of a surrounding area (See Fig. 1). The essential measurement made by a LiDAR sensor is of time, the time that elapses from the moment the pulse is emitted until it returns after being reflected by the target surface. Because the laser pulse travels at the speed of light, time can be directly converted to distance.
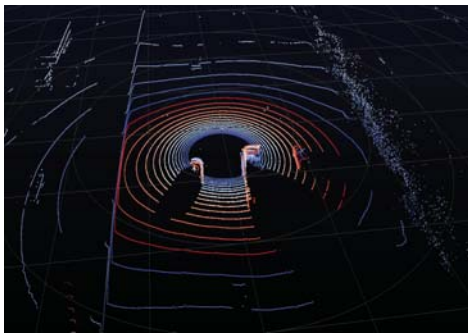

Fig. 1 An example of 3D point cloud map generated by Velodyne HDL-32E LiDAR

The Velodyne HDL-32E LiDAR is an ultra-compact and a more cost-effective version of LiDAR sensor (See Fig. 2). The dimensions of the HDL-32E is $8.5cm \times 8.5cm \times 15cm (L \times W \times H)$ and the net weight is 1.3 kg. It comprises a vertical array of 32 radially-oriented lasers, resulting in an effective $41.3°$ vertical field of view (FOV) from $+10.67°$ to $-30.67°$. The entire unit can spin about its vertical axis at speeds up to 10 Hz to provide a full $360°$ azimuthal field of view. The maximum range of the sensor is 70 meter and it captures approximately 700,000 points per second.

III. PROPOSED METHODOLOGY

Our object is using the direct raw data from LiDAR sensor and provide robust tracking list of potential targets. In our approach, the dense data from sensor is first simplified by segmentation step. Moreover, the points which are beyond the data resolution is removed in the same step. Then clustering algorithm is applied to obtain vehicle-like object. Additionally, data association analysis is assigned to handle measurements in different frames. The detail of these steps are described in the following sections.

A. Spatial Data Structure

A LiDAR dataset is composed of a large number of sparse 3D points. A kd-tree, or k-dimensional tree, is a data structure used in computer science for organizing points in a k-dimensional space [12]. For our purposes we will generally deal with point clouds in three dimensions. kd-tree is very useful for searching nearest neighbor. Considering the number of dimensions k is fixed ($k = 3$), and dataset size is n, we can estimate complexity of the most important operations with kd-tree

- building a kd-tree has $O(nlogn)$ time complexity and $O(kn)$ space complexity
- nearest neighbor search - close to $O(logn)$
- m nearest neighbors - close to $O(mlogn)$

The advantage of kd-tree is that it has been proved the usefulness in the reduction of complexity of already existing three-dimensional models in an automatic and unsupervised way [13].

B. Segmentation

Instead of establishing complicated neighborhood relations and computing cell-based in a grid elevation map [14], In our current method we partition the data in a way such that each single laser can be processed individually. Since the Velodyne HDL-32E has composed of 32 lasers with each laser has a fixed pitch angle, the data from each single laser forms a circle (See Fig. 4 (a)). The diameter of the circle is depending on the height and pitch angle of each laser. Therefore, we can utilize this information as a threshold value in order to separate ground and non-ground points. Considering the symbol S is representing the set of non-ground points. Because the height (H) and the pitch angle (α) are fixed values, so the distance (L_i) is invariant for each single laser by equation

$$L_i = \frac{H}{\alpha} \qquad (1)$$

In other words, if no occlusions occurred during the travel of light between emitter and receiver, the distance from the emitter to the ground is settled in terms of the current slope of the road. So if op_i is less than the settled value L_i, the point p_i is belonging to non-ground point set. As we can see from Fig. 3, the point $p_i \in P$ maps to S if and only if

$$S = \{p_i | \forall p_i \in P, op_i < L_i\} \qquad (2)$$

World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
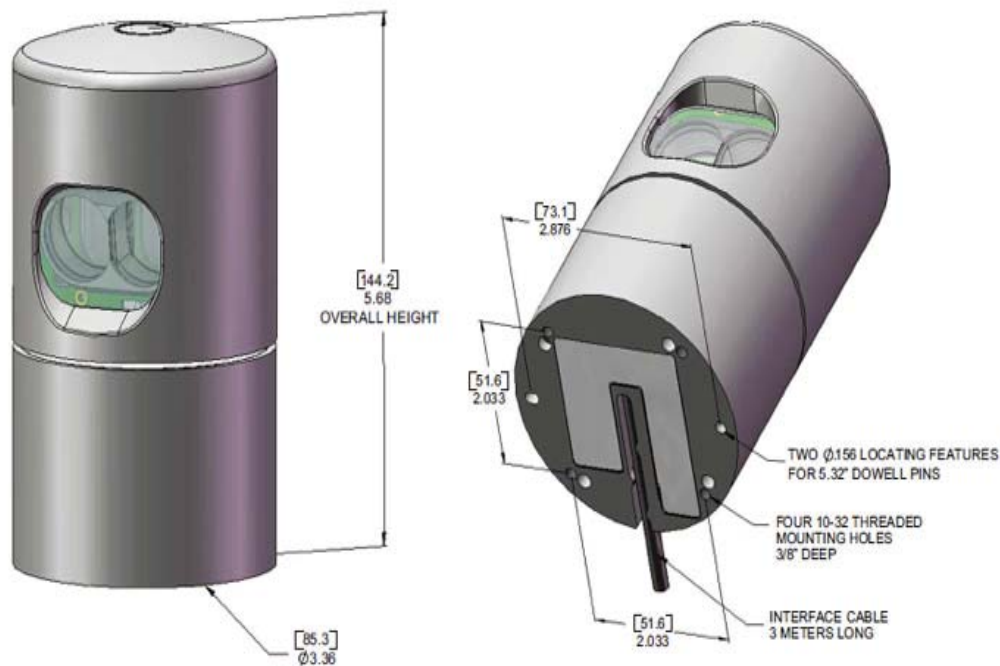Vol:10, No:6, 2016

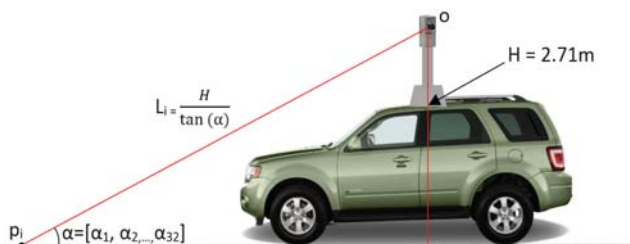Fig. 2 The Velodyne HDL-32E LiDAR Model

Fig. 3 The formula to distinguish non-ground points

By the elevation threshold, we can discriminate the ground point and non-ground point for each single laser. The selected non-ground points are passed through the false alarm mitigation module using a rule based scheme (e.g., minimal road width requirement) to further reduce false alarms. In Fig. 4 (b), the selected non-ground points are indicated by color red. Then our segmentation approach models the ground in every single laser. Thus, we grouped all the selected non-ground point sets from 32 laser data and result in a classified road segmented sets (See Fig. 4 (c)).

C. Clustering

Matching 3D point clouds to geometrical shapes such as planes, cylinders or cube can only succeed if the point clouds are already reasonably segmented. For large 3D point clouds obtained from complicated geometry, graph-based approaches are the most popular class of algorithm for robust and efficient segmentation of 3D laser data, because it can capture arbitrarily shaped clusters. Klaas presents RBNN algorithm by only using the concept of local neighborhood [8]. Our

proposed clustering algorithm is refined version based on this method. Since Klaas's method is non parametric and deterministic. Therefore, the proposed algorithm does not need initialization. The only parameters of RBNN that needed to predefined are the cluster radius and the number of minimum points in one cluster. In RBNN, if neighbor points lies within a predefined radius, the neighbor nodes belongs to the same group. If the number of points in one cluster is less than threshold, the cluster is considered as outliers. One of the advantages of RBNN is that it is dependent on *Euclidean distance*. The algorithm can be described by the following steps

1) scan through all points from the data set.
2) for current point which is not assigned to any cluster

   - search all neighbor points within radius predefined.
   - if any of these neighbors is already assigned to a cluster, mark the current point to the same cluster. Assign the rest of unassigned neighbors to the same cluster.
   - if no neighbor points are assigned, then create a new cluster. Thus assign both current point and neighbors to the new cluster.

3) for current point which is assigned to cluster

   - search all neighbor points within radius predefined.
   - if there exist neighbors assigned to different clusters, merge all these clusters.
   - if no neighbor points are assigned, assign neighbor points to the cluster of current point.

The implementation of RBNN on sample frame data is show at Fig. 5.

World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
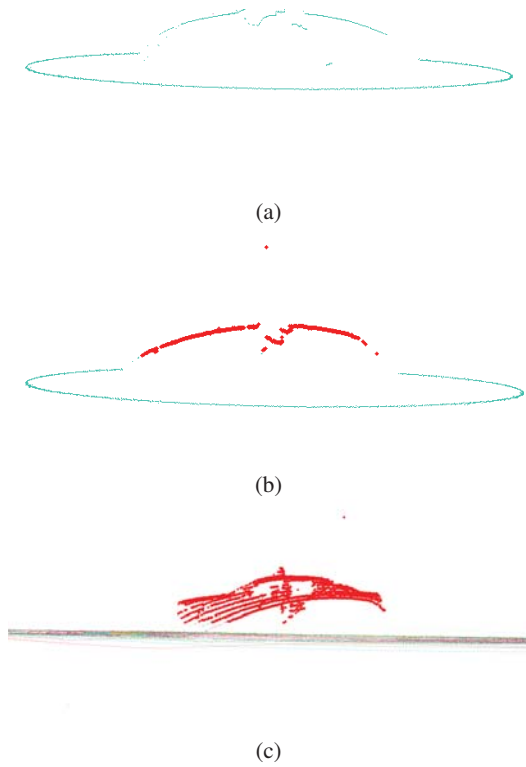Vol:10, No:6, 2016

(a)

(b)

(c)

Fig. 4 (a) Raw point cloud from one single laser. (b) Non-ground points separated from raw data. (c) All non-ground points collected from 32 laser point cloud

D. Data Association and Tracking

The problem of data association is to determine the correspondence between measurements and tracked objects. The data association problem arises at time t when the task is to match the set of tracked objects $T = \{t_1, t_2, ..., t_n\}$ with the set of measurements $O = \{o_1, o_2, ..., o_m\}$ observed in the current frame (See Fig. 6). In the easiest case, the relationship is bijective, which means all objects present are also observed and each measurement was due to a previously tracked object. This is an unrealistic scenario in our application. Surjective or injective mappings that is $n \neq m$ are occurring more often in our application. Surjective associations occur if all measurements can be matched to a previously tracked object $n > m$. In this case, no new objects were detected, but tracked objects may have disappeared. In particular, tracked object t_i may not be visible in frame t because

- object t_i left the LiDARs field of view,
- object t_i is temporarily missing due to a false negative detection by the feature detector,
- object t_i is occluded by an object t_j, which results in a single measurement.

Injective associations occur if all previously tracked objects can be matched to the observations $n < m$. In this case, additional objects may have been detected in the current frame. The additional objects may be new objects entering the LiDARs field of view, or previously tracked objects missing

due to false negative detection or occlusion.

The approach traditionally applied to the data association problem is the Hungarian algorithm, which can be used to find the measurement-to-track mappings in $O(m^3)$ time [15]. The algorithm solves the weighted bipartite graph matching problem. Since all measurements are compared with all active tracks, the method is also called the *global nearest neighbor* (GNN) approach. To realize an estimation position of target at the current frame and further predict the position of target for the next frame, Kalman filter is implemented. The general model is given below:

$$x_{k+1} = A_k u_k + B_\omega \omega_k \qquad (3)$$

$$y_k = C x_k + v_k \qquad (4)$$

where x is the system state vector, y is the measurement vector, u is the input vector, ω is the process noise vector and v is the measurement noise vector. A, B_ω and C are matrices of appropriate dimensions. ω in this case is considered as constant. So, the state equation is given by:

$$x = \begin{bmatrix} x \\ y \\ v_x \\ v_y \end{bmatrix} \qquad (5)$$

which is equivalent to:

$$\begin{bmatrix} x \\ y \\ v_x \\ v_y \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & 0 & dk & 0 \\ 0 & 1 & 0 & dk \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ v_x \\ v_y \end{bmatrix}_k + \begin{bmatrix} w_x \\ w_y \\ w_{v_x} \\ w_{v_y} \end{bmatrix}_k \qquad (6)$$

The control input model which is applied to vector u is given in (11):

$$B = \begin{bmatrix} \frac{dk^2}{2} \\ \frac{dk^2}{2} \\ dk \\ dk \end{bmatrix} \qquad (7)$$

In our application, the control vector u is zero

$$u = 0 \qquad (8)$$

where dk is equal to 0.1 since our LiDAR frame is 10Hz and w_{k-1} is the Gaussian noise of zero mean. Since we only estimate the vehicle position, the measurement matrix is containing only position measurements. Therefore, the measurement matrix, the noise covariance matrix and state noise covariance matrix are given below respectively:

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \qquad (9)$$

$$E_z = \begin{bmatrix} z_x & 0 \\ 0 & z_y \end{bmatrix} \qquad (10)$$

$$E_x = \begin{bmatrix} \frac{dk^4}{4} & 0 & \frac{dk^3}{2} & 0 \\ 0 & \frac{dk^4}{4} & 0 & \frac{dk^3}{2} \\ \frac{dk^3}{2} & 0 & dk^2 & 0 \\ 0 & \frac{dk^3}{2} & 0 & dk^2 \end{bmatrix} \qquad (11)$$

World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
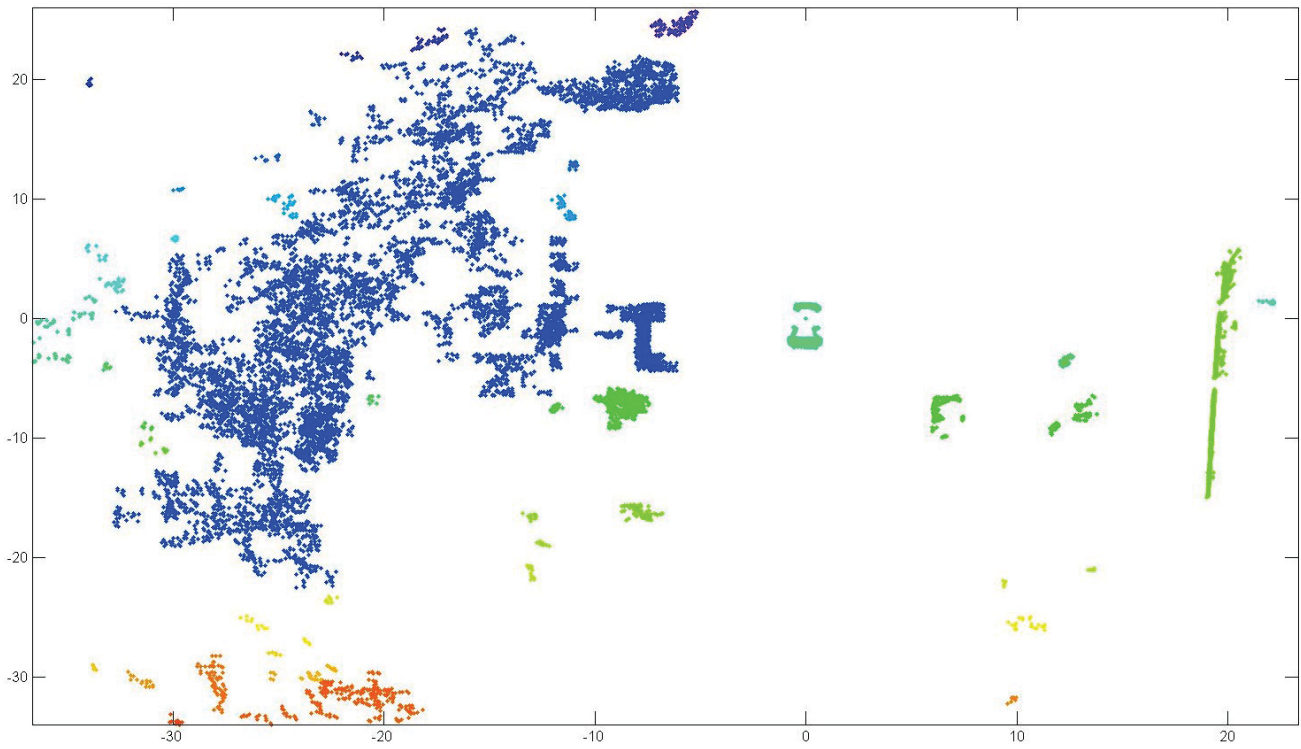Vol:10, No:6, 2016

Fig. 5 RBNN clustering result, running time is 0.118916 seconds
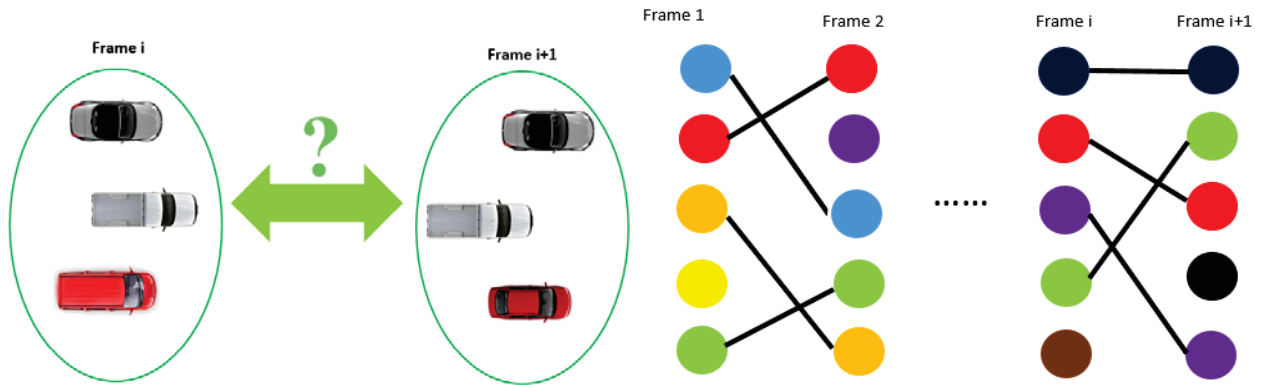

Fig. 6 Data association porblem for multiple target tracking

where the measurement noise in the x axis and y axis is 0.1 and 0.1 respectively. For each set of detections at frame t we first predict the next state of the targets by using the last state estimate and the predicted trajectory:

$$\hat{x}_k = A \cdot x_{k-1} + B \cdot u \qquad (12)$$

Next we predict the next covariance matrix:

$$\hat{P}_k = A \cdot P_{k-1} \cdot A^T + E_x \qquad (13)$$

And Kalman gain is calculated by:

$$K_k = \hat{P}_k \cdot C^T \cdot (C \cdot \hat{P}_k \cdot C^T + E_z)^{-1} \qquad (14)$$

In the standard KF, all the system characteristics have to be specified a priori [16]. However, if there is uncertainty in any of these characteristics, the filter may not be robust enough. For example, the measurement data for KF is processed by classification algorithm that is applied to detect vehicle-like cluster. However, since false positive points are introduced into classified dataset by our algorithms, the measurement noise covariance in KF is uncertainty. We note that the density of cluster that represents vehicles is strong when the vehicle is close to host vehicle, but become extremely weak as the distance between them increases. Consequently, the actual center is variant depends on the distance and direction of the vehicles (See Fig. 7). The blue dot represent the center point of cluster that is being tracked. In fact, the blue point

World Academy of Science, Engineering and Technology
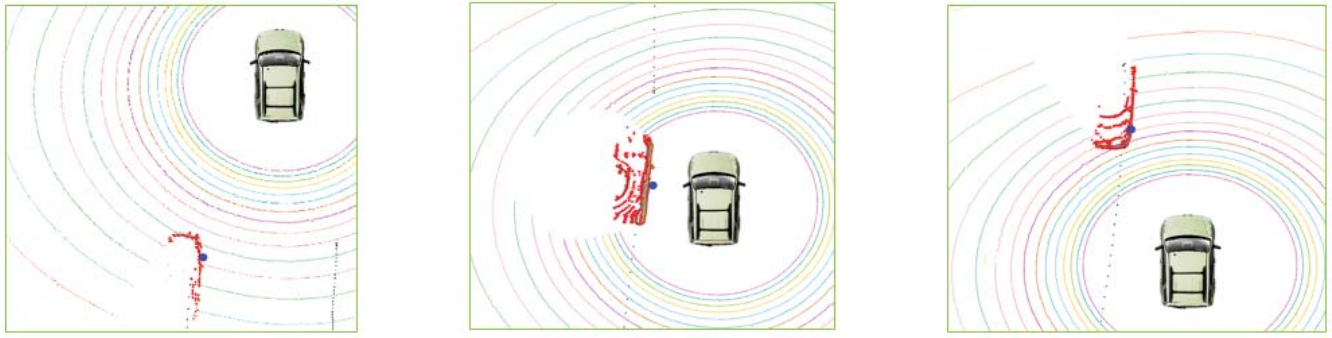International Journal of Computer and Information Engineering
Vol:10, No:6, 2016

Fig. 7 Red color represents detection clusters; Blue color single point represents tracking location; Black color represents trajectories of tracked vehicles
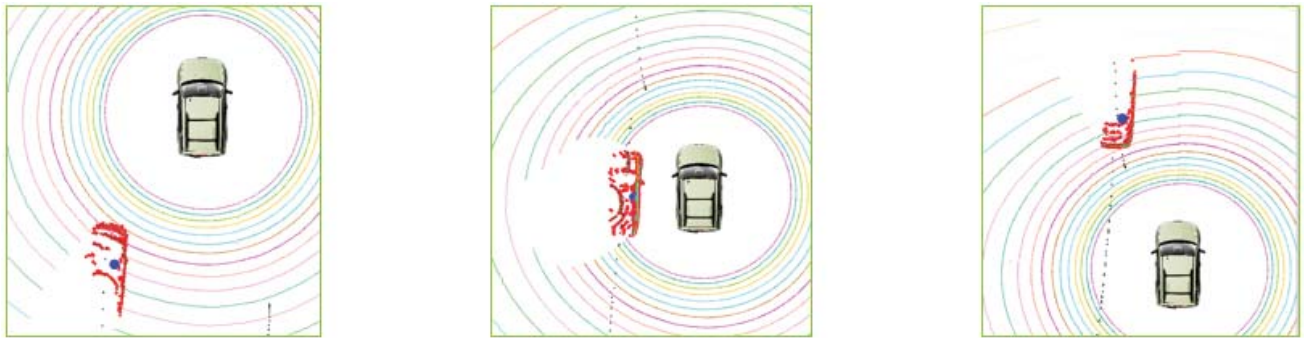

Fig. 8 Covariance based KF result: Red color represents detection clusters; Blue color single point represents tracking location; Black color represents trajectories of tracked vehicles

that is moving from top-right to center-right to bottom-right is depending on where the concentration of the cluster's density. Therefore, to locate the actual center instead of so call data center point is the key challenge for real time tracking.

Instead of following the common practice in vehicle tracking, we will process each vehicle with a general filter to adjust the center point to approximate the actual center point. Thus v is the case that we need to refine. The covariance of v is given as

$$E[v_k v_k^T] = R \qquad (15)$$

Usually, this matrix is a constance value. In our case, as false positive data is increasing as distance increases. Thus an adaptive measurement covariance matrix is proposed in terms of the error propagation. The bigger the value of covariance, the more uncertainty of the measurement. And the maximum constance value for measurement covariance matrix is set to 1. The covariance matrices of the measurement noise are changing from nominal covariances to using scalars R to $\overline{R}$ as follows

$$\overline{R} = 0.00225 * (d_y^2) + 0.1 \qquad (16)$$

where the absolute distance from vehicle to host vehicle in y-axis is d_y. Because the distance is variant from frame to frame, so the measurement covariance matrix value is changing from frame to frame. In addition, the center point is

needed to be adjusted in terms of the actual size of vehicle. Therefore, the offset error equation is

$$e = \gamma * (d_y^2) \qquad (17)$$

where γ is a constant. In this case, we choose $\gamma = \frac{1}{200}$. Combine equation 3.11 and equation 3.12 and update to the KF. The pseudo code is the following

---
Algorithm 1 Covariance Based Kalman Filter

Require: $y_{offset} \geq 0 \wedge y_{offset} \leq 2$
  $d_y \leftarrow$ absolute distance
  $a \leftarrow$ covariance measurement noise matrix coefficient
  $e = (\gamma) * (d_y^2)$
  if $y > 0$ **then**
    $\overline{d_y} = d_y + e$
  else
    $\overline{d_y} = d_y - e$
  end if
  $a = 0.1 + 0.00225 * (\overline{d_y^2})$

TABLE I
COMPARISON OF RESULT

Method	X-axis error(m)	Y-axis error(m)
Normal KF	0.95	1.5
Covariance Based KF	0.45	0.55

World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:10, No:6, 2016

In Fig. 8, the estimated center point (blue dot) is moving to close to real center. From the comparison table, both the error of x-axis and y-axis have been improved by covariance based KF algorithm. Since the size of each sedan is differ from the manufacturer and model. So there is no so called "bench mark" to determine how good the refined algorithm is. However, from the real time 3D point clouds. We can visually see the improvement of the estimated center. Therefore, a combination of the adaptive KF and Hungarian algorithm is introduced. The proposed algorithm consists of the following steps

1) given the positions of measurement objects $O = \{o_1, o_2, ..., o_m\}$ in the current frame t with the set of tracked object's predicted positions $T = \{t_1, t_2, ..., t_n\}$. Form the assignment matrix $A \in \Re^{O_m \times T_n}$ based on Euclidean distance

2) define the auxiliary matrix $Z_{ij} \in \{0, 1\}$, auction the matrix A by satisfying following formulations

$$\text{minimize} \sum_{i=1}^{m} Z_{ij} A_{ij}.$$
$$\text{subject to} \sum_{j=1}^{n} Z_{ij} = 1; \sum_{i=1}^{m} Z_{ij} = 1, \quad (18)$$

3) output the matching score matrix M
4) create or delete the track objects from track list due to either injective or surjective matching, modify the number of track objects n
5) update associated measurements from M to Adaptive KF.
6) process adaptive KF to predict center positions for frame $k+1$
7) repeat from beginning

IV. EXPERIMENTAL RESULT

The experimental data set was collected on QEW highway, Ontario, Canada (see Fig. 10). A Velodyne HDL-32E LiDAR was mounted on the top of the Ford Escape vehicle (see Fig. 3). In order to validate robustness of our proposed algorithm, different scenario were conducted on the highway. The proposed tracking solution was implemented in the MATLAB environment. The ego-vehicle was drove with approximate constant speed at 110km/h.

Fig. 9 (a) presents one scenario of one vehicle at the right lane decelerating and passed by our ego-vehicle. The red color indicates the target is detected and the small green dot means this target is tracked. The black line shows the target's trajectory on the highway. The next scenario showed in Fig. 9 (b) is that a speeding car was detected and tracked when it accelerating on the left lane. Fig. 9 (c) illustrate how the proposed algorithm work for complex situation when multiple vehicles detected and tracked process.

For all the above three scenarios, as the point cloud resolution is decreasing rapidly along the distance. So the perception area is limited to 20m in width and 70m in length in order to reduce the chance of false detection due to the bushes, trees and non regular objects. Consequently, the average process time of our proposed algorithm is about 0.03s compared to 0.1s of LiDAR frame period.
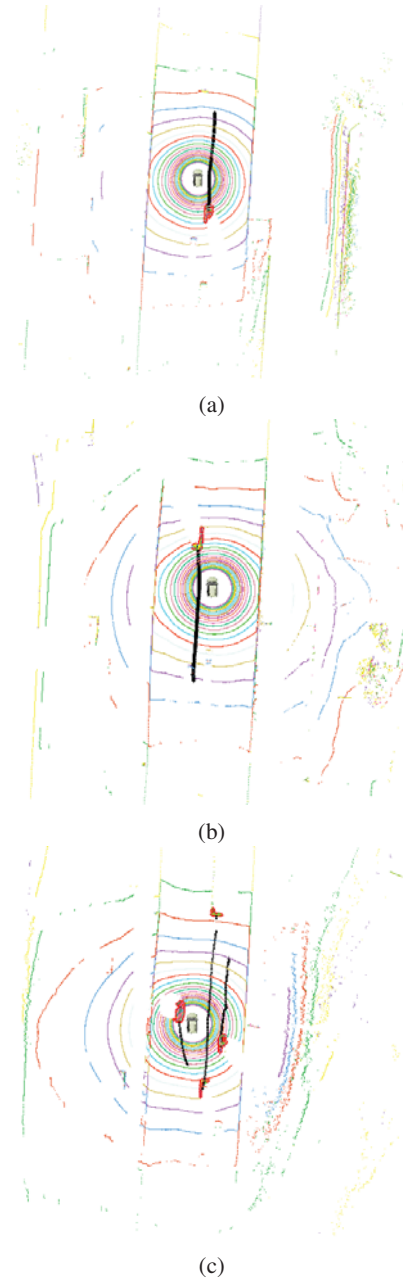

(a)

(b)

(c)

Fig. 9 (a) Slow traffic detected and tracked. (b) Speeding traffic detected and tracked. (c) Multiple vehicles detected and tracked

V. CONCLUSION

In this paper, we proposed a novel real time multiple vehicle detection and tracking algorithm. The algorithm is purely based on a Velodyne HDL-32E LiDAR sensor. The proposed algorithm is processing directly on 3D data, not discarding any important information but allowing for fast and efficient processing. We demonstrated that the proposed algorithm achieves good results on data acquired in expressway environment.

Future work will consist of automatically road boundary detection in order to reduce the number of data points and road surface detection for robust segmentation. In addition, Further

World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:10, No:6, 2016

Fig. 10 Real time implementation

step include complex scenario such as urban environment.

REFERENCES

[1] Manvi Malik and S.Majumder. An integrated computer vision based approach for driving assistance to enhance visibility in all weather conditions. In *International and National Conference on Machines and Mechanisms*, Roorkee,India, December 2013.

[2] Daniel Gohring, Miao Wang, Michael Schnurmacher, and Tinosch Ganjineh. Radar/lidar sensor fusion for car-following on highways. In *Automation, Robotics and Applications (ICARA), 2011 5th International Conference on*, pages 407–412, Wellington, New Zealand, December 2011. IEEE.

[3] Karsten Bohlmann Stefan Laible, Yasir Niaz Khan and Andreas Zell. 3d lidar- and camera-based terrain classification under different lighting conditions. *Autonomous Mobile Systems 2012*, pages 21–29, 2012.

[4] S.Kammel, J.Ziegler, B.Pitzer, M.Werling, T.Gindele, D.Jagzent, J.Schroder, M.Thuy, M.Godel, F.von Hundelshausen, O.Pink, C.Frese, and C.Stiller. Team annieway's autonomous system for the darpa urban challenge 2007. 2008.

[5] M.Montemerlo, J.Becker, S.Bhat, H.Dahlkamp, D.Dolgov, S.Ettinger, D.Haehnel, T.Hilden, G.Hoffmann, B.Huhnke, D.Johnston, S.Klumpp, D.Langer, A.Levandowski, J.Levinson, J.Marcil, D.Orenstein, J.Paefgen, I.Penny, A.Petrovskaya, M.Pflueger, G.Stanek, D.Stavens, A.Vogot, and S.Thrun. Junior: The stanford entry in the urban challenge. 8:569–597, September 2008.

[6] Frank Moosmann, Oliver Pink, and Christoph Stiller. Segmentation of 3d lidar data in non-flat urban environments using a local convexity criterion. In *Intelligent Vehicles Symposium, 2009 IEEE*, pages 215–220, Xi'an,China, June 2009. IEEE.

[7] B.Douillard, J.Underwood, N.Kuntz, V.Vlaskine, A.Quadros, P.Morton, and A.Frenkel. On the segmentation of 3d lidar point clouds. In *Robotics and Automation (ICRA), 2011 IEEE International Conference*, pages 2798–2805, Shanghai,China, May 2011. IEEE.

[8] Klass Klasing, Dirk Wollherr, and Martin Buss. A clustering method for efficient segmentation of 3d laser data. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference*.

[9] Liang Zhang, Qingquan Li, Ming Li, Qingzhou Mao, and Andreas Nuchter. Multiple vehicle-like target tracking based on the velodyne lidar. *IFAC Intelligent Autonomous Vehicles Symposium*, 8:126–131, 2013.

[10] R. E. Kalman. Ba new approach to linear filtering and prediction problems. In *Trans. ASMEVJ. Basic Eng*, pages ser.D, vol. 82, pp.3545, 1960.

[11] Karen Schuckman. Introduction to lasers and lidar. https://www.e-education.psu.edu/geog481/l1_p3.html. Accessed: 2015-10-13.

[12] Andrew W. Moore. *An intoductory tutorial on kd-trees*. Technical Report No. 209, Computer Laboratory,University of Cambridge, 1991.

[13] Daniel Dworak. 3d points cloud reduction using modified k-d tree method. In *VWyjazdowa Sesja Naukowa Doktorantw Politechniki dzkiej*, 2015.

[14] S. Thrun. Stanley: The robot that won the darpa grand challenge. In *Journal of Filed Robot (JFR)*.

[15] M. Betke. Tracking large variable numbers of objects in clutter. In *Computer Vision and Pattern Recognition, CVPR '07. IEEE Conference on*.

[16] Adrian Macaveiu, Andrei Campeanu, and Ioan Nafornita. Kalman-based tracker for multiple radar targets. In *COMM 2014 International Conference on Communications*, Bucharest,Romania, May 2014.

Zhongzhen Luo received the B.Eng. degree in Electrical and Computing Engineering from McMaster University, Hamilton, Ontario, Canada in 2012. He was also working as full time software engineer for over 16 months at ATS automation which is a cambridge, ontario based canadian company that designs and builds factory automation solutions. He is currently working toward the Ph.D degree in Software Engineering at McMaster Automotive Resources Center. His current main research interested include real time implementation for self-driving vehicle, LiDAR point cloud data processing, Information fusion from multiple sensors system and develop and optimize tracking algorithms and filters.

Dr. Saeid Habibi is a Professor and former Chair of the Department of Mechanical Engineering at McMaster University and holds the Senior Industrial Research Chair in Hybrid Technologies sponsored by NSERC and Ford Canada. As founder of the Centre for Mechatronics and Hybrid Technology at McMaster, Dr. Habibi is also co-PI in the 24M Green Auto Powertrain Project funded by the Ontario Research Fund-Research Excellence (ORF-RE) program. His research focuses on intelligent control, state and parameter estimation, fault diagnosis and prediction, variable structure systems (VSS), actuation systems, mechatronics, and fluid power. Application areas include automotive, aerospace, water distribution, and robotics. Dr. Habibi developed the smooth variable structure filter (SVSF) theory, which is a predictor-corrector model-based state estimation strategy that guarantees stability and allows extraction of a higher degree of information from measured signals through secondary indicators of performance. These characteristics make SVSF exceptionally suitable for advanced control as well as for prognostics and health monitoring in automotive systems.

Dr. Martin v. Mohrenschildt is a Professor and former Chair of the Department of Computing and Software Engineering at McMaster University. His research includes model predictive control, hybrid system and Immersive system. Dr. Mohrenschildt also designed and custom built full motion simulator for driving and flying, in which can present accurately timed stimuli (visual, motion, audio, force feedback) to test subjects and measure (EEG, eye-tracking, control response times) their responses.