Efficient L-Shape Fitting for Vehicle Detection Using Laser Scanners

Xiao Zhang, Wenda Xu, Chiyu Dong and John M. Dolan

***Abstract*— The detection of surrounding vehicles is an essential task in autonomous driving, which has been drawing enormous attention recently. When using laser scanners, L-Shape fitting is a key step for model-based vehicle detection and tracking, which requires thorough investigation and comprehensive research. In this paper, we formulate the L-Shape fitting as an optimization problem. An efficient search based method is then proposed to find the optimal solution. Our method does not rely on laser scan sequence information and therefore supports convenient data fusion from multiple laser scanners; it is efficient and involves very few parameters for tuning; the approach is also flexible to suit various fitting demands with different fitting criteria. On-road experiments with production-grade laser scanners have demonstrated the effectiveness and robustness of our approach.**

I. Introduction

Autonomous driving has been studied widely in the literature [1]–[5]. The detection of surrounding objects is an essential task in autonomous driving [6], [7], in which sensing by Light Detection And Ranging (LIDAR) plays an important role. The laser scanner has been widely adopted for the perception of the surrounding environment because the sensor has the capability to measure its distance to the surface of the surrounding objects with high accuracy. A typical way of processing range data is to segment the range data into clusters of points, from which meaningful features such as line segments, chunks, and rectangles are extracted [8]. These features are meaningful because they correspond to objects like walls, bicycles, woods, bushes, vehicles, and pedestrians in the real world [9]. The features are then associated with a static map or tracked targets and used to update the target state through tracking methods such as Multiple Hypothesis Tracking (MHT) [10], [11] or its advanced version which integrates a Rao-Blackwellized Particle Filter (MHT-RBPF) [12], [13].

Even though there is still work to be done before fully autonomous driving arrives, semi-autonomy is already taking place and will be widely introduced in the near future. For example, Adaptive Cruise Control and Collision Warning have already been introduced in high-end luxury vehicles. All these semi-autonomy systems rely heavily on the detection and tracking of moving vehicles, which plays an important role and deserves further study and deeper exploration.

Xiao Zhang, Wenda Xu, and Chiyu Dong are with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, USA {xiaozhan, wendax, chiyud}@andrew.cmu.edu

John M. Dolan is with the Department of Electrical and Computer Engineering and Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA jmd@cs.cmu.edu

Fig. 1: CMU autonomous vehicle research platform "SRX"

In this paper, we focus on L-Shape fitting using laser range data for vehicle tracking. L-Shape fitting is very important, as it provides feeds for vehicle detection and tracking, which is a key component to enable autonomous driving. L-Shape fitting needs to be robust and correct, otherwise it may mislead the object tracking and even cause an accident. L-Shape fitting also needs to be computationally efficient, so that the many high-level estimation and decision tasks depending on it can be completed in real time.

In this paper, we propose a search-based L-Shape fitting approach for vehicle detection from laser data that is computationally efficient. Through on-road experiments with production-grade sensors, the approach is demonstrated to be effective and efficient. The rest of the paper is organized as follows: An overview of related research work is described in Section II. The proposed search-based L-Shape fitting approach is introduced in Section III. Section IV presents the experimental results, from which conclusions are drawn in Section V.

II. Related Work

Carnegie Mellon University's Tartan Racing Team won the DARPA Urban Challenge with its autonomous vehicle "Boss" in 2007. This groundbreaking competition was held in an urban environment with low-density and low-speed traffic, in which the autonomous vehicles were required to perform complex maneuvers such as merging, passing and interacting with both manned and unmanned vehicle traffic. In the competition, the detection of curbs and the tracking of cars were made possible thanks to high-definition 3D LIDAR sensors, e.g. the Velodyne HDL-64, sitting on the top of "Boss". The 3D point clouds provided by such a high-definition 3D LIDAR sensor, together with other sensors in the car, were dense enough to enable "Boss" to clearly understand its surroundings through static map

generation and moving object tracking [14]. However, the high-definition 3D LIDAR sensors are very expensive and still cannot be commonly used on production vehicles.

Carnegie Mellon University's new Cadillac SRX research platform [15] tries to use production-grade sensors, such as LIDAR, RADAR, and camera. These sensors not only provide a potential substitute solution to reduce the equipment expenditure, but also enable a neat appearance for the vehicle that looks familiar to the general public, as these sensors are all concealed in the car (see Fig. 1). However, giving up the high-definition 3D LIDAR sensor and its dense range points presents major challenges for the perception task of the autonomous vehicle. From laser-based range sensing, we can only detect the parts of the object's contour that are facing towards the sensor. Occlusion makes this problem even harder, as the contour of an object may not be fully observed by range sensors. To address these difficulties, a vehicular shape model which abstracts geometric features of the desired target is widely adopted for feature extraction and vehicle pose estimation. With 2D range data, the commonly used vehicular shape models are L shape, boxes, and two perpendicular lines [16], [17]. The features for target tracking can be easily extracted from the vehicular shape model fitted by the incomplete contour.

Based on the vehicular shape model, a variety of fitting methods have been proposed. In [9], Principal Component Analysis (PCA) is applied on range points to extract the most significant axes which potentially correspond to the edges of the car. Comparing poses among successive cycles also helps to find the target's heading direction. In [16], a weighted least-squares method is used to get rid of outliers and fit an incomplete contour to a rectangle model. In consideration of the occlusion problem, both a line fitting and a right-angle corner fitting are conducted in [16], and the corner fit is chosen only when it is significantly better than the line fit. In [17], the ordering of the range points, i.e. the information of the scanning sequence, is utilized to efficiently split the points into two disjoint sets, then two orthogonal lines are fitted by each of the two sets of points respectively, which correspond to the two edges of the car; following the scanning sequence to iterate all these 2-D range points, the proposed algorithm searches for a pivot point, and uses this pivot point to yield those two disjoint sets, i.e., the set of points scanned before the pivot and the set of points scanned after it.

These methods have demonstrated their effectiveness in certain situations, but there are still some limitations. Compared to existing methods, there are three main contributions in this paper. First, our approach does not depend on the laser scanning sequence information and therefore it is easy to achieve data fusion from multiple sensors, which makes it suitable for applications with several production-grade sensors. Secondly, our method is computationally efficient, involving very few parameters and no need for hands-on experience or parameter tuning. Thirdly, our approach is able to accommodate any specified criterion, which makes the approach flexible to suit different fitting demands and extensible for a variety of applications.

Algorithm 1 Adaptive Segmentation Algorithm

Input: range data points $X \in R^{n\times 2}$
Output: set of point clusters S

```
1:  for all x ∈ X do
2:      if x has not been checked then
3:          C ← ∅
4:          Q.push(x)
5:          while Q ≠ ∅ do
6:              x' ← Q.pop()
7:              if x' has not been checked then
8:                  r ← α0||X||
9:                  find all x̄ ∈ X that are within r for x'
10:                 insert all x̄ into C
11:                 mark x' as checked
12:             end if
13:         end while
14:         insert C into S as a new cluster
15:     end if
16: end for
```

III. L-Shape based Vehicle Detection

After obtaining the laser range data, we first segment the data points into clusters. These clusters typically correspond to bicycles, pedestrians, or vehicles and can be classified into each category [9], [18]. In this paper, we are only interested in L-Shape fitting for vehicles. With the assumption of an L-Shape vehicle model, i.e., a rectangle, for each segmented cluster of points, we first search for the best rectangle direction according to a pre-specified criterion, and then obtain the fitted rectangle following that direction and containing all the points in this segmentation.

A. Segmentation

The algorithm for segmentation is displayed in Alg.1, in which the basic idea is to divide the range points into clusters according to a pre-specified distance threshold. The input for the segmentation is the 2-D coordinates of n range scanning points, $X \in R^{n\times 2}$, which are relative to the rear differential of our driverless vehicle. The output of the algorithm is a set of segmented clusters, each of which potentially corresponds to an object in the real world. The main procedure is that for each scan point, we use a K-D tree to find its neighboring points within a distance r and then put them together into a cluster; then we find the points which lie within r distance to any of the points already in the cluster, and again put the newly found points in this cluster; we repeat this process until the cluster does not grow any more, and this finalized cluster serves as one segmentation in the output. This algorithm ensures we only run the range search once for each point.

Note that the segmentation algorithm is adaptive because the range search threshold r is proportional to the distance between the point and the laser sensor (line 8 in Alg. 1). This is justified by the underlying fact that the laser scanning longitude resolution grows with the range distance. In addition, whenever the scan ordering index of the range points is available, the RangeSearch function can use this information to speed up the process. In situations like ours

where scanning sequence is not available, a K-D tree data structure can be used for efficient range searching in this 2-D low-dimensional space.

B. L-Shape Fitting

With the L-Shape rectangle model assumption, for each segmented cluster of points, we want to find a rectangle that fits best to these points. One classical criterion in evaluating the fitting performance is least squares, which involves the following optimization problem:

$$\begin{aligned} \underset{P,\theta,c_1,c_2}{\text{minimize}} \quad & \sum_{i\in P}(x_i\cos\theta + y_i\sin\theta - c_1)^2 \\ & + \sum_{i\in Q}(-x_i\sin\theta + y_i\cos\theta - c_2)^2 \\ \text{subject to} \quad & P\cup Q = \{1,2,...,m\} \\ & c_1, c_2 \in R \quad 0^\circ \le \theta < 90^\circ \end{aligned} \tag{1}$$

in which we minimize the squared error of the corner fitting by looking for the optimal disjunction (P, Q split the m points $\{(x_i, y_i)|i = 1, ..., m\}$ into two sets) and the optimal parameters (θ, c_1, c_2) for the two perpendicular lines which correspond to the points in P and Q respectively - the two line expressions are $x\cos\theta + y\sin\theta = c_1$ and $-x\sin\theta + y\cos\theta = c_2$. However, the above optimization problem turns out to be very difficult to solve due to the combinatorial complexities in the partition problem.

Since accurately solving the above optimization problem is impractical in this real-time application, we instead rely on a search-based algorithm to find the best-fit rectangle approximately. The basic idea is that we iterate through all the possible directions of the rectangle; at each iteration, we can easily find a rectangle oriented in that direction and containing all the scan points; hence the distances of all the points to the rectangle's four edges can be obtained, and based on these distances we can split the points into P and Q, and calculate the corresponding square errors as the objective function in (1); after iterating all the directions and obtaining all the corresponding square errors, we look for the optimal direction which achieves the least squared error, and fit the rectangle based on that direction. Once this fitted rectangle is obtained, the features for vehicle tracking can be easily extracted.

The proposed algorithm is presented in Alg. 2. The input for the algorithm is the m points in this segmentation, $X \in R^{m\times 2}$. The output from this algorithm is the line representations for the four edges of the fitted rectangle. The rectangle's possible direction θ ranges from 0° to 90°, as the two sides of the rectangle are orthogonal and we only care about the single edge that falls between 0° and 90°; the other direction is simply $\theta + \pi/2$. The search space of θ can be reduced if supported with a tracking system or vision scene understanding. If the line representation for the edges is in the form of $ax + by = c$, then the parameters for four edges can be easily obtained by steps 12 to 15 in Alg. 2.

Minimizing the squared error is a commonly used criterion in fitting and regression, but there might also be other criteria

Algorithm 2 Search-Based Rectangle Fitting

Input: range data points $X \in R^{n\times 2}$
Output: rectangle edges $\{a_i x + b_i x = c_i | i = 1, 2, 3, 4\}$
1: $Q \leftarrow \emptyset$
2: **for** $\theta = 0$ **to** $\pi/2 - \delta$ **step** δ **do**
3: $\hat{e}_1 \leftarrow (\cos\theta, \sin\theta)$ ▷ rectangle edge direction vector
4: $\hat{e}_2 \leftarrow (-\sin\theta, \cos\theta)$
5: $C_1 \leftarrow X \cdot \hat{e}_1^T$ ▷ projection on to the edge
6: $C_2 \leftarrow X \cdot \hat{e}_2^T$
7: $q \leftarrow$ CalculatecriterionX(C_1, C_2)
8: insert q into Q with key (θ)
9: **end for**
10: select key (θ^*) from Q with maximum value
11: $C_1^* \leftarrow X \cdot (\cos\theta^*, \sin\theta^*)^T, C_2^* \leftarrow X \cdot (-\sin\theta^*, \cos\theta^*)^T$
12: $a_1 \leftarrow \cos\theta^*, \quad b_1 \leftarrow \sin\theta^*, \ c_1 \leftarrow \min\{C_1^*\}$
13: $a_2 \leftarrow -\sin\theta^*, \quad b_2 \leftarrow \cos\theta^*, \ c_2 \leftarrow \min\{C_2^*\}$
14: $a_3 \leftarrow \cos\theta^*, \quad b_3 \leftarrow \sin\theta^*, \ c_3 \leftarrow \max\{C_1^*\}$
15: $a_4 \leftarrow -\sin\theta^*, \quad b_4 \leftarrow \cos\theta^*, \ c_4 \leftarrow \max\{C_2^*\}$

Algorithm 3 Area Criterion.

1: **function** CalculateArea(C_1, C_2)
2: $c_1^{max} \leftarrow \max\{C_1\}$, $c_1^{min} \leftarrow \min\{C_1\}$
3: $c_2^{max} \leftarrow \max\{C_2\}$, $c_2^{min} \leftarrow \min\{C_2\}$
4: $\alpha \leftarrow -(c_1^{max} - c_1^{min}) \cdot (c_2^{max} - c_2^{min})$
5: **return** α
6: **end function**

suitable in the context of vehicle fitting. The criterion in Alg. 2 is a performance score which reflects how well the rectangle fits with the range points. The criterion can be defined in a variety of ways and each definition might have its strengths and drawbacks. We have considered three criteria for selecting the rectangle: rectangle area minimization, points-to-edges closeness maximization, and points-to-edges squared error minimization, which are presented in Alg. 3, 4, and 5, respectively. Each of the three functions can be chosen to play the role of the CalculatecriterionX function in Alg. 2. All these criterion calculation functions take the input of C_1 and C_2, which are the projections of all the range points on the two orthogonal edges determined by θ.

By using the area minimization criterion in Alg. 3, we are looking for the smallest rectangle that covers all the range points; this is similar to the fitting method used in [19]. By using Alg. 4, we emphasize how close these range points are to the two edges of the right corner. In the projected 2-D plane, c_1^{max} and c_1^{min} specify the boundaries on axis $\hat{e}_1$ for all the points, and the vectors $c_1^{max} - C_1$ and $C_1 - c_1^{min}$ record all the points' distances to the two boundaries; from the two boundaries we choose one that is closer to the range points, and denote the corresponding distance vector as D_1; the distance vector D_2 is defined similarly for the projection axis $\hat{e}_2$. The closeness score is defined as $\sum_{i=1,...,m} 1/d_i$, with d_i being the i-th point's distance to its closest edge. In this way, both decreasing the distance and increasing the number of scanned points would lead to a higher score. Note that there's a minimum distance threshold d_0 to avoid (1) division by zero for points on the boundary, and (2) dramatic voting power for points very close to the edges. By using

Algorithm 4 Closeness Criterion.

Parameter: d_0

function CalculateCloseness(C_1, C_2)
  $c_1^{max} \leftarrow \max\{C_1\}$, $c_1^{min} \leftarrow \min\{C_1\}$
  $c_2^{max} \leftarrow \max\{C_2\}$, $c_2^{min} \leftarrow \min\{C_2\}$
  $D_1 \leftarrow \arg\min_{v \in \{c_1^{max} - C_1, C_1 - c_1^{min}\}} ||v||_{l_2}$
  $D_2 \leftarrow \arg\min_{v \in \{c_2^{max} - C_2, C_2 - c_2^{min}\}} ||v||_{l_2}$
  $\beta \leftarrow 0$
  for $i = 1$ **to** length(D_1) **step** 1 **do**
    $d \leftarrow \max\{\min\{D_{1(i)}, D_{2(i)}\}, d_0\}$
    $\beta \leftarrow \beta + 1/d$
  end for
  return β
end function

Algorithm 5 Variance Criterion.

function CalculateVariance(C_1, C_2)
  $c_1^{max} \leftarrow \max\{C_1\}$, $c_1^{min} \leftarrow \min\{C_1\}$
  $c_2^{max} \leftarrow \max\{C_2\}$, $c_2^{min} \leftarrow \min\{C_2\}$
  $D_1 \leftarrow \arg\min_{v \in \{c_1^{max} - C_1, C_1 - c_1^{min}\}} ||v||_{l_2}$
  $D_2 \leftarrow \arg\min_{v \in \{c_2^{max} - C_2, C_2 - c_2^{min}\}} ||v||_{l_2}$
  $E_1 \leftarrow \{D_{1(i)} | D_{1(i)} < D_{2(i)}\}$
  $E_2 \leftarrow \{D_{2(i)} | D_{2(i)} < D_{1(i)}\}$
  $\gamma \leftarrow -$ variance$\{E_1\} -$ variance$\{E_2\}$
  return γ
end function

Alg. 5, we emphasize the squared error of the two orthogonal edges fitted by the two disjoint groups of points. E_1 contains the to-boundary distances of points that are assigned to $\hat{e}_1$, while E_2 contains the distances for points assigned to $\hat{e}_2$. The variance of E_1 is actually equivalent to the square error of a line in direction θ to the points belonging to E_1. Since in calculating the square error, we are essentially calculating the variance of these distances, this criterion is also termed 'variance'. Recall that by using this variance criterion, we are actually looking for an approximated solution to the optimization problem in (1).

IV. Experimental Results

Experimental results are presented in this section to evaluate the correctness and efficiency of our proposed algorithms. The experiments were tested on CMU's autonomous vehicle SRX (Fig. 1) on local roads. Six IBEOs are mounted which provide multiple layers of range scan. Note that the scanning sequence/ordering information is not used for the experiments here.

A. Computation Efficiency

The computational efficiency of the algorithm is demonstrated through experiments with approximately 10,000 laser scans collected on local roads. Each laser scan is segmented into clusters (around 25,000 clusters in total) and the search-based fitting algorithm is carried out for each cluster. The computation time is presented in Table I. The algorithms are implemented in MATLAB and run on a Linux laptop equipped with an Intel Core i7 CPU. The variance minimization criterion is the most time-consuming because of the heavy computation in calculating the variance, which on average takes around 3.84 ms per cluster. The computation performance of the algorithms will be even better if implemented in a more efficient programming language. Note that the standard deviation of the computation time is very small, which is beneficial as this indicates a consistent estimate of the computation time for high-level decision tasks.

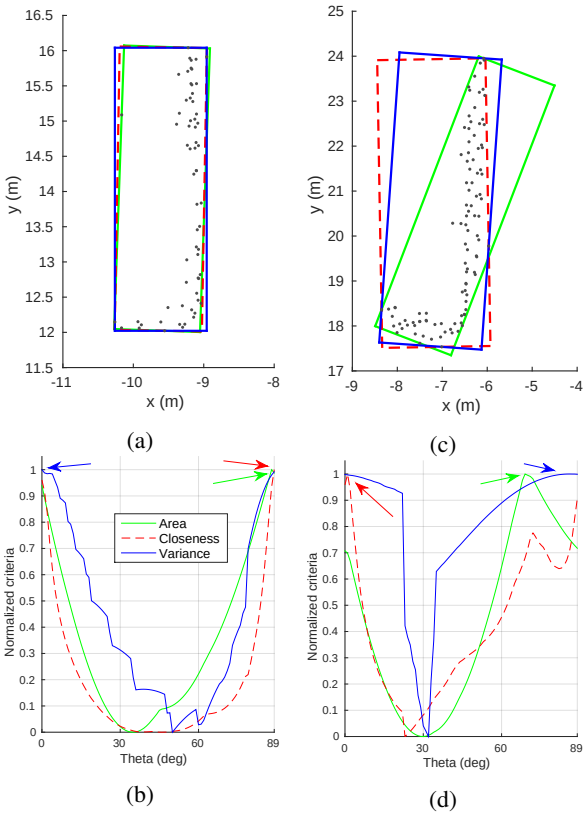

Fig. 2: Rectangle fitting examples. In (a) and (c), the grey dots represent the laser range scan points and the rectangles in green, red, and blue are the fitted rectangles by using criteria area minimization, closeness maximization, and variance minimization. The normalized scores for the three criteria over the searched directions are plotted in (b) and (d), respectively. In example (a), the fitting results from the three criteria are very similar, and the maxima of the three curves in (b) are very close (marked by arrows and achieved at 88°, 89°, and 0°, respectively). In example (b), the fitting result from the area criteria is different from the other two, and its maximum in (d) is away from the other two (achieved at 69°, 1°, and 86°, respectively).

TABLE I: Computation Time of Rectangle Fitting

Criterion	Mean (ms)	STD (ms)
Area Minimization	3.56	0.19
Closeness Maximization	4.00	0.23
Variance Minimization	8.37	0.32

B. Rectangle Fitting

Typical rectangle fitting examples using the three criteria are displayed in Fig. 2. In Fig. 2(a), the fitted rectangles by the three criteria are almost the same; the corresponding normalized criteria along the searched directions are displayed

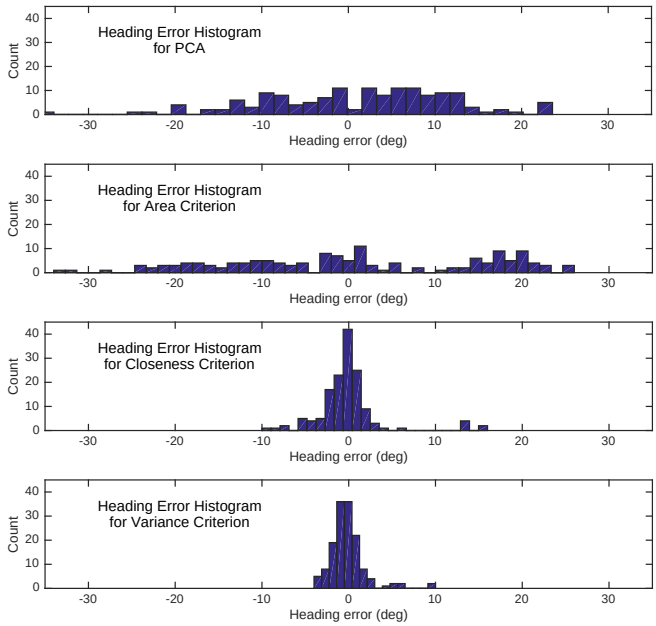

Fig. 3: Histogram of heading error.

TABLE II: Heading Error in Rectangle Fitting

	Real Error ($\theta - \theta_g$)		Absolute Error ($\|\theta - \theta_g\|$)	
Method	Mean (deg)	STD (deg)	Mean (deg)	STD (deg)
PCA	-5.60	10.68	10.09	6.57
Area	0.65	14.80	11.78	8.46
Closeness	0.01	3.65	2.47	3.36
Variance	-0.15	2.19	1.55	1.66

in Fig. 2(b), in which the maxima of the three curves are achieved at nearby θs. In Fig. 2(c), the area minimization criterion yields an inaccurate rectangle heading while the other two criteria are almost the same. A dataset consisting of 145 clusters of points, i.e., the laser range points for 145 vehicles, whose headings are manually labeled at resolution 1°, is used to test the correctness of the proposed approach. The closeness maximization criterion and the variance minimization both have very high correctness, while the area minimization criterion sometimes fails in getting the right heading. The performance of the area minimization criterion is only guaranteed when the range scan point density in the segmentation is high, e.g. in situations of fitting a small object or when a more powerful sensor such as the Velodyne is used [19]. Histograms of heading error using the three criteria and PCA are displayed in Fig. 3, in which the heading error is defined as the heading yielded by our approach minus the heading of ground truth. The mean and standard deviation for both the real error (i.e. heading error) and the absolute error are listed in Table II, in which the mean real error is an indication of estimation bias, while the mean absolute error reflects the estimation accuracy (average magnitude of the errors without considering their direction). As shown in Table II, both the real error and absolute error are small for

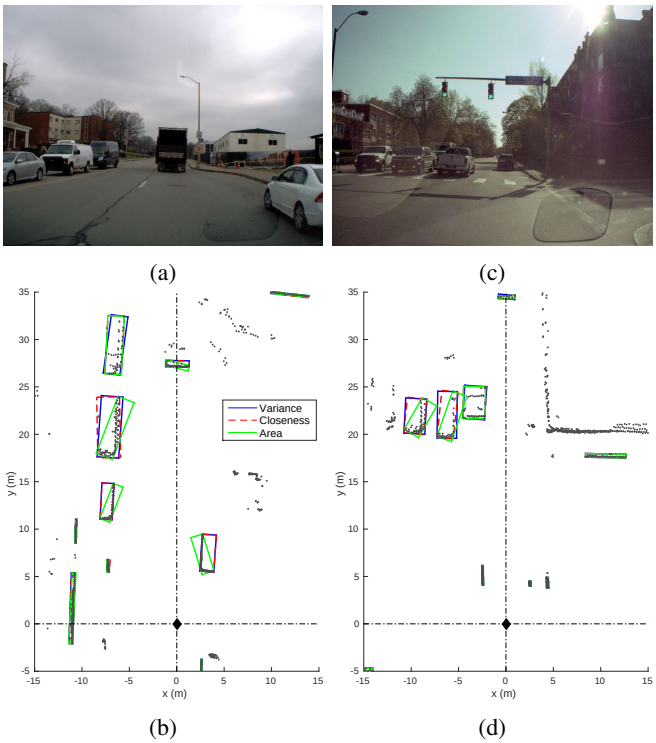

Fig. 4: Segmentation and vehicle fitting results for typical laser scan cycles. The camera views for these two cycles are displayed in (a) and (c). In (b) and (d), the segmented laser scan points are represented by markers with varying colors and shapes, the vehicle fitting results by criterion closeness and variance are displayed by rectangles in red and blue, respectively.

the closeness and the variance criteria.

The results from two typical cycles are displayed in Fig. 4. The five cars scattered around in Fig. 4(a) are very well fitted in Fig. 4(b), and the three cars crossing the intersection in Fig. 4(c) are very well fitted in Fig. 4(d). The position of our car is denoted by the filled black diamond at coordinates (0, 0). Note that small clusters with fewer than ten range points are ignored in the implementation.

Fig. 5 gives examples of the rare cases when the methods did not achieve good performance. In Fig. 5(b), the closeness criterion (rectangle in red) misunderstood the vehicle heading due to the points outside of the L-Shape, which come from the scan points of the white SUV's side mirror in Fig. 5(a); in Fig. 5(e), the variance criterion was jeopardized by the points scattered around the lower part of the vehicle, and these points arise from the back window of the truck. The underlying reason for the imperfection here is that the assumed L-Shape vehicle model does not hold for the two examples, either due to the side mirror outside of the L-Shape or because of the back window inside of the L-Shape. Note that these imperfect cases are very rare, and their impact on the vehicle tracking is very limited, since the fitting results have always been corrected in their successive cycles, as demonstrated by Fig. 5(c) and Fig. 5(d). It is also the case that in these rare situations, when one of the two criteria is jeopardized, the other one gives the correct rectangle fitting.

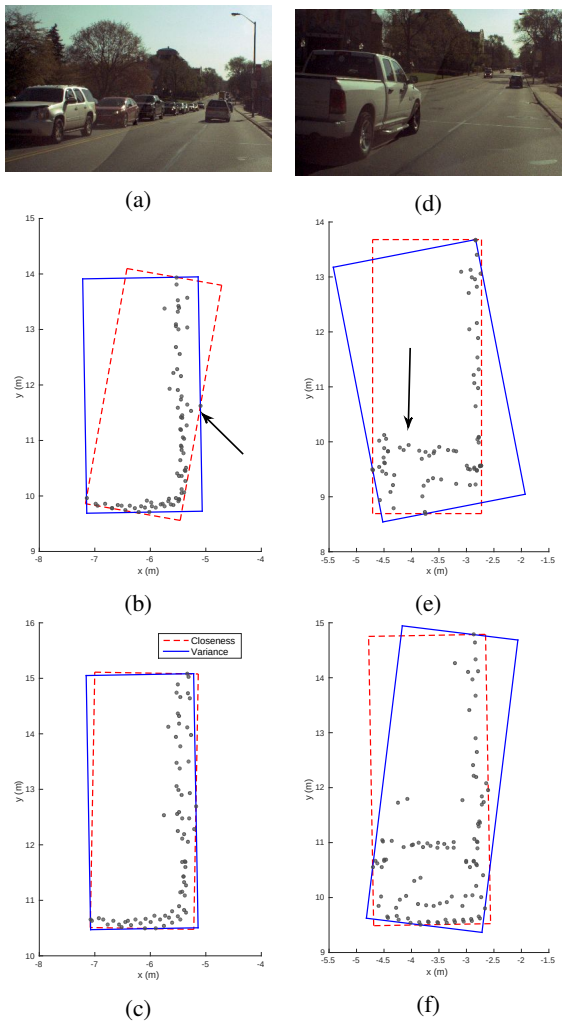

Fig. 5: Cases when the L-Shape model assumption does not hold. For the white SUV in (a), the segmented laser scan points and the rectangle fitting results in two successive cycles are displayed in (b) and (c). For the truck in (d), the scan points and the fitting results in two successive cycles are displayed in (e) and (f). In (b), the closeness criterion in red is jeopardized by the side mirror as marked by the black arrow. In (d), the variance criterion in blue is influenced by the truck's back window. Note that even for these two rare situations at least one of the two criterion works well and the fitting results get corrected in the next cycle.

V. CONCLUSIONS

A search-based L-Shape fitting approach for laser range data is proposed in this paper. The proposed approach does not depend on the scan sequence/ordering information, enabling fusion of raw laser data from multiple laser scanners; the approach is computationally efficient and easy to implement, involving very few parameters and no need for hands-on experience or parameter tuning; it is able to accommodate any specified criterion, which makes the approach flexible for different fitting demands and extensible for a variety of applications. Three criteria have been discussed in the paper and compared in the experiments. As demonstrated by the on-road experiments with production-grade sensors, the proposed approach is effective and robust even under circumstances when the L-Shape model assumption does not hold, e.g., the laser scan of a SUV with big side mirror. The mean absolute error in vehicle heading estimation can be as low as $1.55°$. In our future work, we will incorporate the proposed L-Shape fitting approach into a general vehicle tracking algorithm.

REFERENCES

[1] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann *et al.*, "Stanley: The robot that won the DARPA Grand Challenge," *Journal of field Robotics*, vol. 23, no. 9, pp. 661–692, 2006.

[2] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer *et al.*, "Autonomous driving in urban environments: Boss and the Urban Challenge," *Journal of Field Robotics*, vol. 25, no. 8, pp. 425–466, 2008.

[3] M. Althoff, O. Stursberg, and M. Buss, "Model-based probabilistic collision detection in autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 2, pp. 299–310, 2009.

[4] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt *et al.*, "Towards fully autonomous driving: Systems and algorithms," in *IEEE Intelligent Vehicles Symposium*, 2011, pp. 163–168.

[5] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the KITTI vision benchmark suite," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3354–3361.

[6] I. Miller, M. Campbell, and D. Huttenlocher, "Efficient unbiased tracking of multiple dynamic obstacles under large viewpoint changes," *IEEE Transactions on Robotics*, vol. 27, no. 1, pp. 29–46, 2011.

[7] W. Xu, J. Snider, J. Wei, and J. M. Dolan, "Context-aware tracking of moving objects for distance keeping," in *IEEE Intelligent Vehicles Symposium*, 2015, pp. 1380–1385.

[8] C. Mertz, L. E. Navarro-Serment, R. MacLachlan, P. Rybski, A. Steinfeld, A. Suppe, C. Urmson, N. Vandapel, M. Hebert, C. Thorpe *et al.*, "Moving object detection with laser scanners," *Journal of Field Robotics*, vol. 30, no. 1, pp. 17–43, 2013.

[9] H. Zhao, Q. Zhang, M. Chiba, R. Shibasaki, J. Cui, and H. Zha, "Moving object classification using horizontal laser scan data," in *IEEE International Conference on Robotics and Automation*, 2009, pp. 2424–2430.

[10] S. S. Blackman, "Multiple hypothesis tracking for multiple target tracking," *IEEE Aerospace and Electronic Systems Magazine*, vol. 19, no. 1, pp. 5–18, 2004.

[11] I. J. Cox and S. L. Hingorani, "An efficient implementation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 2, pp. 138–150, 1996.

[12] D. Schulz, D. Fox, and J. Hightower, "People tracking with anonymous and id-sensors using rao-blackwellised particle filters," in *Proc. of the International Joint Conference on Artificial Intelligence*, 2003, pp. 921–928.

[13] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE transactions on Robotics*, vol. 23, no. 1, pp. 34–46, 2007.

[14] M. S. Darms, P. E. Rybski, C. Baker, and C. Urmson, "Obstacle detection and tracking for the urban challenge," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 3, pp. 475–485, 2009.

[15] J. Wei, J. M. Snider, J. Kim, J. M. Dolan, R. Rajkumar, and B. Litkouhi, "Towards a viable autonomous driving research platform," in *IEEE Intelligent Vehicles Symposium*, 2013, pp. 763–770.

[16] R. MacLachlan and C. Mertz, "Tracking of moving objects from a moving vehicle using a scanning laser rangefinder," in *IEEE Intelligent Transportation Systems Conference*, 2006, pp. 301–306.

[17] X. Shen, S. Pendleton, and M. H. Ang, "Efficient L-shape fitting of laser scanner data for vehicle pose estimation," in *IEEE Conference on Robotics, Automation and Mechatronics*, 2015, pp. 173–178.

[18] M. Sonka, V. Hlavac, and R. Boyle, *Image processing, analysis, and machine vision*. Cengage Learning, 2014.

[19] C. Urmson, J. A. Bagnell, C. R. Baker, M. Hebert, A. Kelly, R. Rajkumar, P. E. Rybski, S. Scherer, R. Simmons, S. Singh *et al.*, "Tartan racing: A multi-modal approach to the DARPA Urban Challenge," 2007.