Open access • Book Chapter • DOI:10.1007/978-3-642-00196-3_10

Efficient Techniques for Dynamic Vehicle Detection — Source link

Anna Petrovskaya, Sebastian Thrun

Institutions: Stanford University

Published on: 01 Jan 2009 - International Symposium on Experimental Robotics

Topics: Vehicle tracking system

Efficient Techniques for Dynamic Vehicle Detection

Anna Petrovskaya and Sebastian Thrun

Computer Science Department
Stanford University
Stanford, California 94305, USA
{ anya, thrun }@cs.stanford.edu

Summary. Fast detection of moving vehicles is crucial for safe autonomous urban driving. We present the vehicle detection algorithm developed for our entry in the Urban Grand Challenge, an autonomous driving race organized by the U.S. Government in 2007. The algorithm provides reliable detection of moving vehicles from a high-speed moving platform using laser range finders. We present the notion of motion evidence, which allows us to overcome the low signal-to-noise ratio that arises during rapid detection of moving vehicles in noisy urban environments. We also present and evaluate an array of optimization techniques that enable accurate detection in real time. Experimental results show empirical validation on data from the most challenging situations presented at the Urban Grand Challenge as well as other urban settings.

1 Introduction

Self-driving cars promise to bring a number of benefits to society, including prevention of road accidents, optimal fuel usage, comfort and convenience. In recent years the U.S. Government has organized a series of competitions for autonomous vehicles in order to encourage research in this area. In 2005, autonomous vehicles were able to complete a 131 mile course in the desert. In the 2007 competition, the Urban Grand Challenge (UGC), robots were presented with an even more difficult task: autonomous navigation in urban environments. In this competition the robots had to drive safely with respect to other robots, human-driven vehicles and the environment. They also had to obey the rules of the road as described in the California rulebook. One of the most significant changes from the previous competition is the necessity for situational awareness in an urban setting, including both static and dynamic parts of the environment. In this paper we describe dynamic vehicle detection techniques we developed for our robot Junior (pictured in Fig. 1). Junior demonstrated safe driving skills and won the second place in the 2007 Urban Grand Challenge competition. We are concerned with detection of moving

(a) (b)

Fig. 1. (a) Our robot Junior (blue) negotiates an intersection with human-driven vehicles at the qualification event for the Urban Grand Challenge in November 2007. (b) Junior, is equipped with five different laser measurement systems, a multi-radar assembly, and a multi-signal inertial navigation system.

vehicles from a high-speed mobile platform (the ego-vehicle) using laser range finders. Detection of vehicles is usually discussed as a sub-problem in vehicle tracking literature [1, 2, 3, 4]. However, fast and accurate detection of new vehicles is more challenging and more computationally expensive than tracking of existing targets.

In this paper we focus on the vehicle detection sub-problem alone. A detailed description of the full vehicle tracking module is given in [5]. For autonomous driving, fast detection of new moving vehicles is crucial in order to avoid dangerous situations and possible collisions. Poor signal-to-noise ratio presents a significant obstacle to fast detection of new moving vehicles. We present the notion of *motion evidence* that allows us to quickly and accurately detect new vehicles by effectively pruning false positives caused by noise. We also present an array of optimization techniques that assures reliable real time performance in the challenging traffic conditions, including situations presented at the Urban Grand Challenge. In the experimental section we evaluate the impact of each technique on the overall performance.

The rest of this paper is organized as follows. The next section provides the necessary background and discusses state-of-the-art in vehicle detection and tracking. Section 3 introduces notation and describes our models of vehicles, sensor data, and measurements. Section 4 describes our vehicle detection algorithm, motion evidence notion and optimization techniques. Experimental results are given in Sect. 5. We conclude with a discussion in Sect. 6.

2 Background

Since vehicle detection is a prerequisite for vehicle tracking, it is often described in the vehicle tracking literature [1, 2, 3, 4]. Typically vehicle tracking approaches proceed in three stages: data segmentation, data association, and Bayesian filter update. During data segmentation the sensor data is divided into meaningful pieces (usually lines or clusters). During data association these

pieces are assigned to tracked vehicles. Next a Bayesian filter update is performed to fit targets to the data.

The second stage - data association - is generally considered the most challenging stage of the vehicle detection and tracking problem because of the association ambiguities that arise. Typically this stage is carried out using variants of multiple hypothesis tracking (MHT) algorithm (e.g. [2, 3]). The filter update is usually carried out using variants of Kalman filter (KF), which is augmented by interacting multiple model method in some cases ([1, 3]).

Although vehicle tracking literature primarily relies on variants of KF, there is also a great body of multiple target tracking literature for other applications (see [6] for a summary) where parametric, sample-based, and hybrid filters are used. For example [7] uses a Rao-Blackwellized particle filter for multiple target tracking on simulated data. A popular alternative to MHT for data association is the joint probabilistic data association (JPDA) method. For example in [8] a JPDA particle filter is used to track multiple targets from an indoor mobile robot platform.

We utilize a model based approach, which uses particle filters and eliminates the need for separate data segmentation and association stages. We described this approach in [5], where we used it to solve the full tracking problem via a Rao-Blackwellized particle filter that estimated position, velocity and shape of tracked vehicles.

The main focus of this paper is on techniques for fast and accurate moving vehicle detection. In the prior art, the detection problem has been solved by addition of vision sensors (e.g. [4]), although visual classification does not help distinguish moving vehicles from stationary. Another approach is to sample frames at lower rates to overcome the low signal-to-noise ratio ([3]), although it increases the time it takes to detect a new moving vehicle. Other described approaches detect by scan shape ([1, 2]) or by location ([3]). Due to possible ambiguities in the range data, these approaches tend to have lower detection accuracy.

3 Representation

3.1 Vehicle model

We detect each vehicle using a separate particle filter. For each vehicle we estimate its 2D position and orientation $X_t = (x_t, y_t, \theta_t)$ in world coordinates at time t, and its forward velocity v_t. Figure 2 depicts a dynamic Bayes network representing the resulting probabilistic model. We assume that the velocity evolves from one time step to the next by addition of random bounded acceleration. Furthermore we utilize a linear motion model for vehicle dynamics: first the orientation is perturbed slightly, then the vehicle moves forward according to its velocity, then the orientation is perturbed slightly again. This

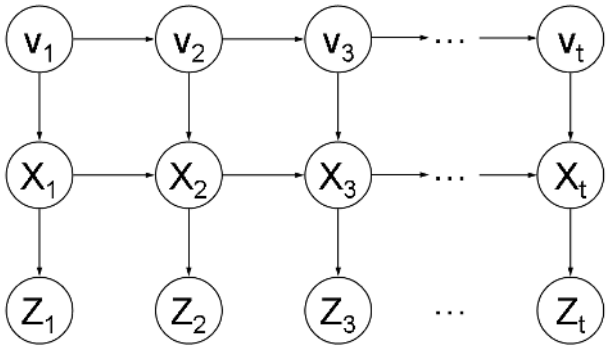


Fig. 2. Dynamic Bayesian network model of the detected vehicle pose X_t, forward velocity v_t, and measurements Z_t.

motion law is often utilized when exact dynamics of the object are unknown, which happens to be the case in our application.

The exact geometric shape of a vehicle can be complex and difficult to model precisely. For simplicity we approximate it by a rectangular shape of a fixed width W and length L[1]. The 2D representation is sufficient because the height of the vehicles is not important for driving applications.

At each time step we obtain a new measurement Z_t. The measurements are incorporated into the particle filter according to the measurement model that we describe in Sect. 3.3.

3.2 Sensor data representation

We use two types of laser range finders for sensing the environment: IBEO Alasca and Velodyne HDL-64E. These sensors produce range scans at 10Hz. IBEO produces four horizontal scan lines and performs ground filtering. Velodyne produces unfiltered 3D point clouds consisting of 64 horizontal scan lines.

2D data sets are more compact than 3D point clouds. Furthermore they are sufficient for vehicle tracking, provided that ground readings can be filtered out. To expedite data access, we pre-process the raw data and filter out ground readings to build 2D *virtual scans* as described in [5]. A virtual scan is a grid in polar coordinates, which subdivides 360° around a chosen origin point into angular grids (Fig. 3). In each angular grid we record the range to the closest obstacle. We will often refer to the cone of an angular grid from the origin until the recorded range as a *ray* due to its similarity to a laser ray.

By construction each angular grid contains information about free, occupied, and occluded space. This information is important for detecting changes

[1] The width and the length of each vehicle can also be estimated as we have shown in [5].

in the environment. The changes are computed by differencing two consecutive virtual scans. This computation takes time linear in the size of the virtual scan and only needs to be carried out once per frame. Figure 3 shows the results of a virtual scan differencing operation. The classification of space into free, occupied and occluded also helps us properly reason about what parts of a vehicle should be visible as we describe in Sect. 3.3.

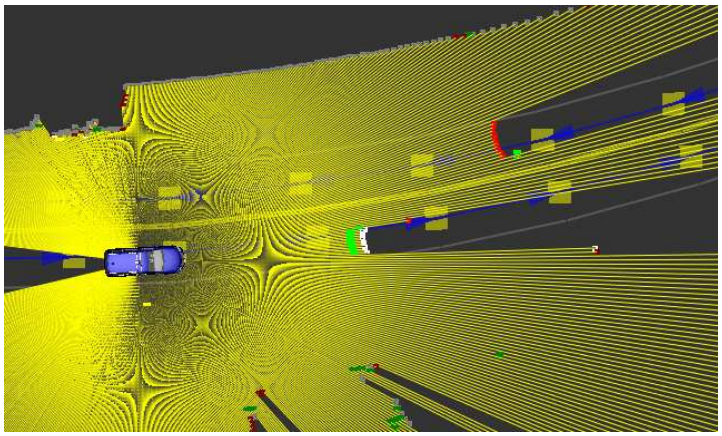


Fig. 3. A virtual scan constructed from Velodyne data. Yellow line segments represent virtual rays. Red points are new obstacles, green points are obstacles that disappeared, and white points are obstacles that remained unchanged or appeared in previously occluded areas. (Best viewed in color.)

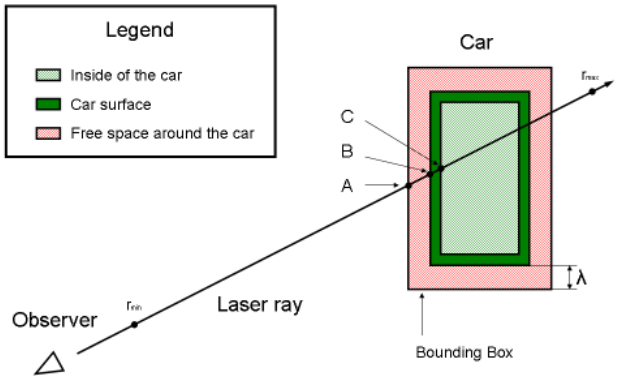
3.3 Measurement model

To complete our probabilistic model, we define the following measurement model[2]. Given a vehicle's pose X and a virtual scan Z we need to compute the measurement likelihood $p(Z|X)$. We position a rectangular shape representing the vehicle according to X. Then we build a bounding box to include all points within a predefined[3] distance λ around the vehicle (see Fig. 4). Assuming that there is an actual vehicle in this configuration, we would expect that points within the rectangle to be occupied or occluded, and points in its vicinity to be free or occluded, because vehicles are spatially separated from other objects in the environment.

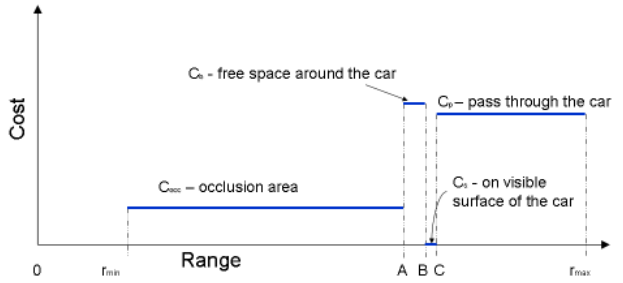
We consider measurements obtained along each ray independent of each other (a common assumption when dealing with laser range finders). Thus if we have a total of N rays in the virtual scan Z, the measurement likelihood factors as follows:

$$p(Z|G, X) = \prod_{i=1}^{N} p(z_i|G, X).$$

[2] We utilize the same measurement model as described in [5].
[3] We used the setting of $\lambda = 1m$ in our implementation.



(a)



(b)

Fig. 4. Measurement likelihood computations. (a) shows the geometric regions involved in the likelihood computations. (b) shows the costs assignment for a single ray. (Best viewed in color.)

We model each ray's likelihood as a zero-mean Gaussian of variance σ_i^2 computed with respect to a cost c_i selected based on the relationship between the ray and the vehicle (η_i is a normalization constant):

$$P(z_i|G,X) = \eta_i \ \exp\{ -\frac{c_i^2}{\sigma_i^2} \}.$$

The costs and variances are set to constants that depend on the region in which the reading falls into (see Fig. 4 for illustration). c_{occ}, σ_{occ} are the settings for range readings that fall short of the bounding box and thus represent situations when another object is occluding the vehicle. c_b and σ_b are the settings for range readings that fall short of the vehicle but inside of the bounding box. c_s and σ_s are the settings for readings on the vehicle's visible surface (that we assume to be of non-zero depth). c_p, σ_p are used for rays that extend beyond the vehicle's surface.

The domain for each range reading is between minimum range r_{min} and maximum range r_{max} of the sensor. Since the costs we select are piece-wise

constant, it is easy to integrate the unnormalized likelihoods to obtain the normalization constants η_i. Note that for the rays that do not target the vehicle or the bounding box, the above logic automatically yields uniform distributions as these rays never hit the bounding box.

Note that the above measurement model naturally handles partially occluded objects including objects that are "split up" by occlusion into several point clusters.

4 Vehicle detection

Accurate moving vehicle detection in laser range data requires three frames. The first two frames are required to detect motion of an object. The third frame is required to check that the motion is consistent over time and follows the vehicle dynamics law. Thus for a 10Hz sensor the minimum vehicle detection time is 0.3 seconds.

Note that detection based on three frames allows for accurate results, because we can observe two consecutive motion updates and verify that the observed motion is consistent with a moving vehicle. For some applications it may be acceptable to sacrifice accuracy in favor of faster detection based on just one or two frames. For example in [3] objects that appear in areas previously seen as empty are detected as "moving". Often this approach is adopted when it is desired to filter out moving obstacles to build a static map.

4.1 The basic algorithm

Our vehicle detection method proceeds in three stages:

1. First a vehicle is fitted using importance sampling in an area where a change in the environment has been detected by scan differencing. The scoring is performed using the measurement model described in Sect. 3.3.
2. Next the vehicle's velocity is estimated by performing a particle filter update step and scoring using the measurement model in the next frame.
3. During the last stage, another particle filter update is performed and scored against a third frame.

4.2 Challenges in vehicle detection

The range data in outdoor urban environments contains large amounts of noise that adds up from a number of sources. The laser range finder produces readings corrupted by random noise. Environmental factors such as dust and rain cause false readings. Complex shape of vehicles does not match the box models precisely. Readings obtained from the same object at a slightly different height give different range.

For the driving application we need to detect vehicles moving at 5mph to 35mph with a 10Hz sensor. Thus a vehicle moves $20 - 150$cm per frame. This signal can be easily overwhelmed by noise especially in the lower range of the velocities. The poor signal-to-noise ratio makes it difficult to accurately tell a moving object apart from noise in just three frames.

Although the signal is easier to detect if we use more than three frames, this solution is undesirable because it increases the detection time and takes up more computational resources. A more efficient approach, proposed in [3], is to sample the frames at a lower rate (e.g. 1Hz), so that the signal is prevalent over the noise. However, this method also increases the total time required for detection of a vehicle and therefore it is unsuitable for our application.

4.3 Motion evidence

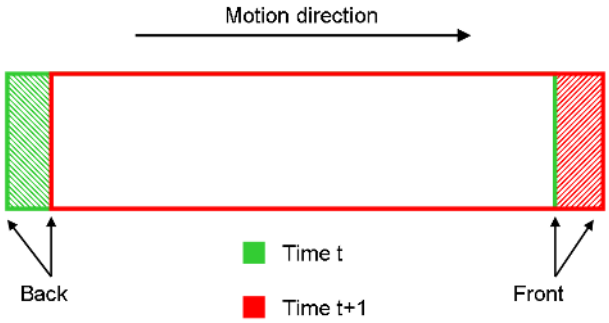


Fig. 5. Diagram representing forward motion of a bus. Green color represents the position of the bus at time t. Red color represents its position at time $t + 1$. The green shaded area in the back of the bus frees up as the bus moves forward. The red shaded area in the front of the bus becomes occupied. Note that these changes are small compared to the overall area taken up by the bus, which remains occupied in both frames. (Best viewed in color.)

To overcome the poor signal-to-noise ratio, we turn to the method used by humans to detect moving vehicles in noisy data. Consider a long bus moving forward at 5mph (Fig. 5). From one frame to the next it travels 20cm - a negligible distance compared to the noise and overall size of the vehicle. Since the middle of the bus appears stationary, a human trying to discern motion will focus on the front and back of the bus, to see if there is at least a tad of motion.

To take advantage of the same method for vehicle detection, we define a score we call *motion evidence*. To compute this score, we consider the regions cleared by the vehicle as it moves. The cleared area behind the vehicle should be occupied in the prior frame and free in the current frame. Similarly the area in front of the moving vehicle should be free in the prior frame and occupied

in the current frame. Usually we can only observe the front or the back of the vehicle, thus only half of the evidence is available due to self-occlusion.

Note that motion evidence score is different from the probabilities obtained by fitting a vehicle using a particle filter. The particle filter computes the probability that motion *could have* happened, whereas the motion evidence scores the motion that *"must have"* happened. In the bus example given above the motion evidence score would ignore the entire bus except 20cm in the front and in the back.

The motion evidence score can be computed for any pair of consecutive frames. In our approach we compute it for the first and the second pairs of frames. Doing so provides a very dramatic decrease in false positives, without affecting the false negatives rate.

4.4 Optimizations

New vehicle detection is the most challenging and computationally expensive part of tracking dynamic vehicles. Below we describe the optimization techniques we developed to achieve reliable vehicle detection in real time. In Sect. 5 we evaluate the impact of each technique on the performance of vehicle detection.

Road masking

Since a digital road map is available in our application, one simple optimization is to restrict the search to the road regions. We do this by marking each data point as "close to road" or "far from road". Only the points near the road are considered for new vehicle detection. This optimization greatly improves the efficiency of the vehicle detection algorithm.

Cleared area

As we already discussed above, a change in the data can be caused by either noise or motion. Ultimately the motion evidence score will help disambiguate motion from noise. However, the motion evidence score can only be used after the vehicle model has already been fitted to data. To make the search more efficient we would like to distinguish between noise and motion before performing any model fittings.

When a vehicle moves forward with a minimum velocity v_{min} for a time interval Δt, it clears an area of approximately $v_{min} \quad \Delta t \quad W$. Thus we can examine each data point to see if enough space has been cleared around it to allow for motion of a vehicle. If the vehicle is moving away from us, the cleared area will be in the current frame with respect to the prior frame. If the vehicle is approaching us, the cleared area will be in the prior frame with respect to the current frame. Thus we can find both types of cleared area by performing a symmetric clearing operation between the two frames.

Even though cleared area logic is not as powerful as the motion evidence score, it provides a significant speed-up when used as a fast data pre-processing step.

Scaling Series

The first step of vehicle detection involves fitting the geometric vehicle model to a virtual scan under conditions of large uncertainty: several meters in position and 360° in orientation of the vehicle. Using simple importance sampling with three state parameters makes the problem intractable within real time constraints.

To improve performance we turn to Scaling Series, a method first proposed in [9] for a tactile localization application. In that application the number of parameters was also too large to perform an importance sampling step in real time in conditions of global uncertainty. They proposed the Scaling Series algorithm to efficiently produce a much more informed proposal distribution, one that is concentrated around the areas of high probability mass. We refer the reader to [9] for details on Scaling Series, but briefly, the algorithm works by performing a series of successive refinements, generating an increasingly informative proposal distribution at each step of the series. The successive refinements are performed by gradually annealing the measurement model from artificially relaxed to realistic.

In our setting, we applied the Scaling Series algorithm to choose the proposal distribution for the initial importance sampling step. Using this method we obtained a very significant improvement in the reliability of the search and reduced the time it takes to detect a new moving vehicle by a factor of 10.

Backward search

Since vehicle detection takes three frames, the minimum detection time is 0.3 seconds for a sensor with a frame rate of 10Hz. It turns out that if we only search forward in time, then the minimum detection time is 0.4 seconds for approaching vehicles, because the first frame is only used to detect dynamic data points in the second frame. However, if we fit the vehicle in the second frame and then move it backwards in time, we can utilize the first frame as well. In this case we use frame number two for the initial vehicle fitting and frame number one for velocity estimation. As before the third frame is used to check motion consistency.

5 Experimental Results

The presented vehicle detection algorithm has been used as part of our tracking module (described in [5]). The module has proven to be reliable, efficient,

Fig. 6. Actual scene and tracking results on a test presented in Area A during the qualification event at the Urban Grand Challenge. During this test robots had to repeatedly merge into live traffic on a course resembling the Greek letter θ.

and capable of handling complex traffic situations including the most challenging tests presented during the Urban Grand Challenge and the qualifiers (see Fig. 6). The average computation time of our approach - including detection and tracking - is 25ms per frame, which is four times faster than the sensor update rate.

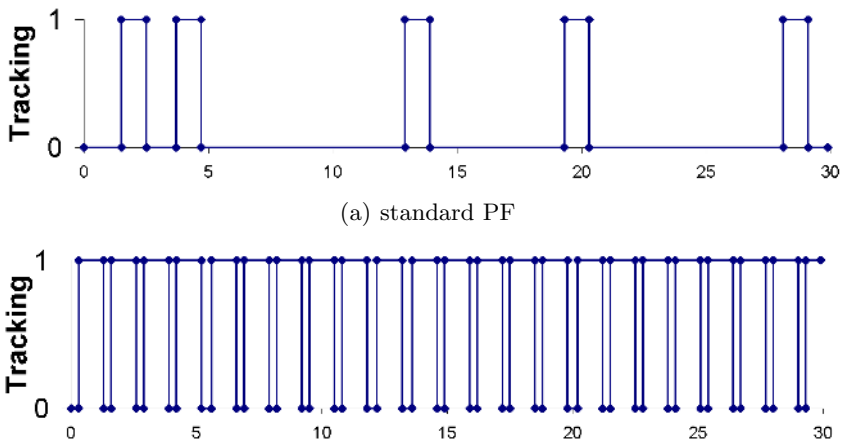
	Total	Detected in Frame			False	% Detected by Frame			FP
Data sets	Cars	3	4	5	Detections	3	4	5	%
Area A	713	596	103	14	1	83.6	98.0	100.0	0.1
Stanford	679	645	32	2	2	95.0	99.7	100.0	0.3
Alameda	532	485	45	2	5	91.2	99.6	100.0	0.9
Overall	1,924	1,726	180	18	8	89.7	99.1	100.0	0.4

Table 1. Vehicle detector performance on data sets from three urban environments. For each car we counted how many frames it took to detect it. By construction of the algorithm, at least three frames are required. We also counted the number of false detections. The '% Detected' columns give the percentages of cars detected by frame three, four and five. 'FP %' is the false positive rate attained by the vehicle detection algorithm.

To evaluate the performance of the vehicle detection algorithm empirically we forced the tracking module to drop each target as soon as it was detected. We then ran vehicle detection on data sets from three different urban environments: Area A of the Urban Grand Challenge qualifiers, the Stanford campus, and a port town in Alameda, CA (see Tbl. 1). In each frame of data we labeled all vehicles identifiable by a human in the range data. The vehicles had to be within 50m of Junior, on or near the road and moving with a speed of at least 5mph. For each vehicle we counted how many frames it took to detect it. We also counted false positives. Overall, all vehicles were detected in five frames or less and the false positive rate was 0.4%.

To evaluate motion evidence contribution, we ran the algorithm with and without motion evidence logic on labeled data sets. The use of motion evidence brought false discovery rate from 60% down to 0.4%. At the same time the rate of false negatives did not increase.

We used prerecorded data sets to evaluate performance gains from the optimization techniques. We compared the computation time of the algorithm with and without road masking. Road masking sped up the algorithm by a



(a) standard PF

(b) Scaling Series

Fig. 7. Comparison of standard PF to Scaling Series for new vehicle detection. The horizontal axis denotes time in seconds. The vertical axis has two states: 0 - target is not tracked, 1 - target is tracked. To verify target acquisition, the code was specifically modified to discontinue tracking a target after 1 second. By construction of the algorithm, the minimum possible time spent in non-tracking state is 0.3 seconds. (a) standard PF has a long target acquisition time - too dangerous for autonomous driving. (b) Scaling Series method has nearly perfect acquisition time.

factor of eight. We also ran the algorithm with and without cleared area logic. The speed up from this optimization was approximately a factor of three. The backward search optimization reduced the minimum detection delay for oncoming traffic by 25%.

To evaluate improvements from Scaling Series, we used a 30 second data set of our ego-vehicle following another car. For evaluation purposes we modified the tracker to drop each target after tracking it for 1 second. Figure 7 presents comparison of results obtained using a standard particle filter and Scaling Series particle filter. Vehicle detection with the standard particle filter took 4.44 seconds on average and 13.7 seconds in the worst case, which can easily result in a collision in a real life situation. In contrast the Scaling Series particle filter took 0.32 seconds on average to detect the vehicle, with the worst case being 0.5 seconds. Thus the Scaling Series approach performs very close to the theoretical minimum of 0.3 seconds.

Several videos of vehicle detection and tracking using the techniques presented in this paper are available at the website

http://cs.stanford.edu/∼anya/uc.html

6 Conclusions

We presented a model based approach to detection of dynamic vehicles from a high-speed robotic platform equipped with laser range finders. We developed the notion of motion evidence, which effectively overcomes the low signal-to-noise ratio for fast and accurate detection of moving vehicles in noisy urban environments. We also presented an array of optimization techniques that enable our algorithm to run in real time and provide reliable moving vehicle detection even in the most challenging conditions presented at the UGC.

A promising direction for future work is to fuse laser range finders with other sensors such as vision and radar to allow for even faster detection of new vehicles. Another useful direction is to identify a greater variety of moving objects in urban settings such as people, bicyclists and animals.

Acknowledgment

This research has been conducted for the Stanford Racing Team and would have been impossible without the whole team's efforts to build the hardware and software that makes up the team's robot Junior. The authors thank all team members for their hard work. The Stanford Racing Team is indebted to DARPA for creating the UGC, and for its financial support under the Track A Program. Further, Stanford University thanks its various sponsors.

References

1. L. Zhao and C. Thorpe. Qualitative and quantitative car tracking from a range image sequence. In *Computer Vision and Pattern Recognition*, 1998.
2. D. Streller, K. Furstenberg, and K. Dietmayer. Vehicle and object models for robust tracking in traffic scenes using laser range images. In *Intelligent Transportation Systems*, 2002.
3. C. Wang, C. Thorpe, S. Thrun, M. Hebert, and H. Durrant-Whyte. Simultaneous localization, mapping and moving object tracking. *The International Journal of Robotics Research, Sep 2007; vol. 26: pp. 889-916*, 2007.
4. S. Wender and K. Dietmayer. 3d vehicle detection using a laser scanner and a video camera. In *6th European Congress on ITS, Aalborg, Denmark*, 2007.
5. A. Petrovskaya and S. Thrun. Model based vehicle tracking for autonomous driving in urban environments. In *RSS, Zurich, Switzerland*, 2008.
6. S. S. Blackman. Multiple hypothesis tracking for multiple target tracking. *IEEE AE Systems Magazine*, 2004.
7. S. Särkkä, A. Vehtari, and J. Lampinen. Rao-blackwellized particle filter for multiple target tracking. *Inf. Fusion*, 8(1), 2007.
8. D. Schulz, W. Burgard, D. Fox, and A. Cremers. Tracking multiple moving targets with a mobile robot using particle filters and statistical data association. In *ICRA*, 2001.
9. A. Petrovskaya, O. Khatib, S. Thrun, and A. Y. Ng. Bayesian estimation for autonomous object manipulation based on tactile sensors. In *ICRA*, 2006.