Stable and Efficient Tracking of Multiple Dynamic Obstacles Under Large Viewpoint Changes

Isaac Miller, *Student Member, IEEE,* Mark Campbell, *Member, IEEE,* and Dan Huttenlocher, *Member, IEEE*

***Abstract*—The LocalMap tracking algorithm is presented as a computationally feasible, real-time solution to the joint estimation problem of data assignment and dynamic obstacle tracking from a potentially moving robotic platform. The algorithm utilizes a Bayesian factorization to separate the joint estimation problem into 1) a data assignment problem solved via particle filter, and 2) a multiple dynamic obstacle tracking problem solved by efficient parametric filters developed specifically for tracking full-size vehicles in a dense traffic environment. The algorithm is validated in controlled experiments with full-size robotic vehicles, and on data collected at the 2007 DARPA Urban Challenge, where it was used in real-time.**

***Index Terms*—Detection and tracking of moving obstacles, Rao-Blackwellized particle filter.**

I. Introduction

MODERN autonomous mobile robots operating at human scales of size face the significant challenge of understanding the dynamic environment in which they (and we) interact. The challenge is more than a pedagogical curiosity, as failures in a human-populated operating environment could turn science fiction into nightmarish science fact. These dangers are particularly acute for full-size robotic vehicles, where tracking accidents and traffic accidents are separated only by a thin layer of hardware.

The primary difficulty lies not in sensor deficiency, but in sensor interpretation. Since no single sensor directly measures positions and velocities of all moving objects in the environment, some form of data fusion is required to build an understanding of the environment incrementally over time. Techniques for doing so have evolved in several active areas of research, from target tracking for missile and aerospace defense to collision warning systems for automotive driver assistance [1], [2]. Among the various approaches adopted in these fields, two central problems are commonly addressed. The first is measurement assignment, where sensor data is divided and distributed among objects to account for the fact that no sensor perfectly segments each object in the environment. The second is feature extraction and tracking, where raw sensor data is preprocessed into detections of specific object features to be used as inputs in tracking algorithms. This latter problem is typically addressed to reduce large amounts of raw sensor data into a small set of organized metadata.

I. Miller and M. Campbell are with the Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY, 14853 USA e-mail: {itm2, mc288}@cornell.edu.

D. Huttenlocher is with the Computer Science Department, Cornell University, Ithaca, NY, 14853 USA e-mail: dph@cs.cornell.edu.

Manuscript received MMM DD, YYYY; revised MMM DD, YYYY.

Several successful methods have arisen in the target tracking literature for addressing the data assignment problem [1]. The simplest is *maximum likelihood* or *nearest neighbor* data assignment, where measurements are assigned to the objects according to a best-fit criterion. Many popular robotic simultaneous localization and mapping (SLAM) and detection and tracking of moving obstacles (DATMO) algorithms rely on this method of data assignment [3]–[5]. Although simple to implement, these approaches tend to make assignment mistakes in situations where objects are temporarily indistinguishable, such as when two moving objects cross paths. More accurate approaches acknowledge such situations, either by assigning measurements to more than one object in the case of the *joint probabilistic data association filter*, or by maintaining multiple assignment hypotheses in the case of *multiple hypothesis tracking* [1], [6], [7]. A recent class of approaches to gain momentum are *Monte Carlo data association* techniques, whereby Monte Carlo methods are used to randomly generate likely data assignments or complete tracking hypotheses [8]–[10].

On the other side of the problem, research in automotive driver assistance has yielded feature extraction and tracking algorithms suitable for full-size automotive environments. Many of the feature extraction algorithms have been designed specifically to operate on clouds of points generated by automotive laser rangefinders, including circle detection for pedestrian leg recognition, center of mass measurements for vehicle tracking, rectangle and bounding box fitting, corner detection, and line and segment detection for vehicle identification [2], [11]–[18]. Other automotive algorithms combine features from complementary sensors, such as laser rangefinders and optical cameras, to detect and track vehicles [19]–[22].

From a robotics point of view, correct understanding of the dynamic environment requires simultaneous solutions for both the data assignment and the feature extraction and tracking problems. The challenge lies in the fact that these two problems are inextricably linked in a single joint estimation task, because no sensor can perfectly identify and separate all the objects in a populated dynamic environment. Many of the aforementioned automotive feature extraction and tracking algorithms ignore this fact for computational reasons, as it is combinatorially expensive to track multiple objects from detections of their respective parts. While such an assumption makes the automotive tracking problem more tractable, the instability of feature extraction algorithms yields erroneous tracking artifacts in most realistic automotive environments [17]. In contrast, the works of Maehlisch *et al.* specifically address the joint estimation problem with a *probabilistic*

hypothesis density filter, which fuses an optical camera with a small laser rangefinder to simultaneously estimate the number of moving objects and their trajectories [19], [21]. Computationally, the expense of this and other integrated multitarget tracking algorithms limits tracking to an environment with a small number of targets, where most sensor data is rejected as clutter.

This paper presents the LocalMap algorithm, a practical multitarget tracking algorithm meant to approach data assignment and dynamic obstacle tracking as a single joint estimation problem under bearable computational load. The LocalMap algorithm extends the previous work of Miller and Campbell, in which simulations of a similar obstacle tracking algorithm were shown capable of simultaneously determining the number of obstacles in an environment and tracking those obstacles as they moved [23]. Preliminary experimental results for stable obstacle tracking with a single stationary laser rangefinder were also presented. Here, the same joint estimation techniques are extended to stable, accurate, real-time multisensor tracking of a large number of potentially dynamic obstacles from a potentially dynamic robot. Specifically, the LocalMap algorithm abandons commonly-used parameterized models of target geometry in favor of a richer point cloud representation and more stable measurement updates. The new representation yields accurate estimation in two practical but very difficult scenarios: when one or more dynamic obstacles occupy a large portion of sensor fields of view, and when the dynamic obstacles' observed geometry changes rapidly due to maneuvers at short sensor ranges. The LocalMap algorithm is derived to fuse three common types of full-size robotic sensors: laser rangefinders, radars, and optical cameras, though it is general to any mode of sensing. The capabilities of the LocalMap algorithm are verified with controlled field experiments conducted from full-size vehicles traveling at city speeds. Experimental results are also given for Cornell University's entry in the DARPA Urban Challenge, a 6 hour, 60 mile urban robotics challenge for which the LocalMap algorithm was implemented. Section II introduces the LocalMap algorithm, providing a theoretical foundation for the data assignment and obstacle tracking joint estimation problem. Section III presents experimental performance for the LocalMap algorithm, first in controlled full-size vehicle tracking experiments, and then in an uncontrolled environment full of traffic. Section IV summarizes the performance of the LocalMap algorithm in the DARPA Urban Challenge, and Section V concludes.

II. The LocalMap Tracking Algorithm

Derivation of the LocalMap algorithm begins by casting the dynamic urban perception problem in a Bayesian framework. To that end, a set of variables O describing obstacle positions and motions are estimated to permit basic robotic navigation and obstacle avoidance. In this context, the cardinality of the obstacle variables O implicitly represents the number of obstacles in the world, and may change over time. In addition to the obstacle variables O, a set of data assignment variables A are also estimated. These assignment variables record the history of sensor measurements generated by each obstacle, thereby dividing the user robot's sensor measurements into historical sequences associated with each individual obstacle. Neither the obstacle variables O nor the assignment variables A are known with certainty in the unstructured environment, so both must be estimated. The LocalMap algorithm therefore estimates the joint probability density over these two sets of variables, conditioned on all available sensor measurements:

$$p\left(A_k, O_k | Z_k\right) \tag{1}$$

where Z_k are the set of observed sensor measurements, and the subscript k represents an integer time index. The use of capital letters A_k, O_k, and Z_k indicates a full time history of these quantities, from the filter's inception at time index 0 to the present time index k.

Though general filtering methods exist to estimate joint probability densities over multiple variables, the case of perception of a dynamic urban environment on a mobile robot presents several difficulties. First, the probability density in equation 1 is hybrid, because the data assignment variables A are discrete while the obstacle variables O are continuous. Second, the number of potential data assignment histories grows exponentially in time, number of measurements, and number of obstacles, preventing exact probabilistic reasoning even in modest environments. A final challenge is the fact that obstacles moving in a dynamic urban environment may be intelligent, capable of executing complex nonlinear maneuvers over time. These three challenges in concert make most traditional estimation approaches unsuitable for estimating the joint probability density in equation 1.

To help make the estimation problem tractable amidst these challenges, the joint probability density is first factorized. The definition of conditional probability is employed to rewrite the joint probability density in equation 1:

$$p\left(A_k, O_k | Z_k\right) = p\left(A_k | Z_k\right) \cdot p\left(O_k | Z_k, A_k\right) \tag{2}$$

This factorization is exact, and is similar to one made by Montemerlo *et al.* to estimate the joint density between an ego robot's pose and the positions of static landmarks in a variant of the SLAM problem [24]. Intuitively, the factorization made in equation 2 is beneficial because it decouples the discrete data assignment estimation problem $p\left(A_k | Z_k\right)$ from the continuous tracking problem $p\left(O_k | Z_k, A_k\right)$. Although the discrete data assignment problem is still too large to be solved with exact inference techniques, the continuous tracking problem conditioned on known data assignments may be solved with less expensive parametric filters such as the Kalman Filter (KF) or Sigma Point Filter (SPF) [1], [25], [26].

A. The Discrete Data Assignment Problem

Despite the factorization made in equation 2, the number of discrete data assignment permutations still grows exponentially in time, number of obstacles, and number of measurements. It is therefore computationally infeasible to evaluate the corresponding data assignment density $p\left(A_k | Z_k\right)$ exactly, so approximation techniques are used instead. For the LocalMap algorithm, the discrete data assignment density $p\left(A_k | Z_k\right)$ is

approximated by a small number of randomly-drawn samples, utilizing well-worn Monte Carlo likelihood-weighted sampling techniques [27], [28]. The goal of the factorization made in equation 2 is to make these non-deterministic data assignment choices 'obvious,' reducing sample size requirements through the use of intelligent but inexpensive parametric solutions to the continuous tracking problem.

This type of hybrid particle filter, which estimates a factorized probability density by combining Monte Carlo sampling techniques with closed-form parametric filters, is known as a Rao-Blackwellized Particle Filter (RBPF); see, for example, Doucet *et al.* [29]. Särkkä *et al.* have studied such an estimator in the context of target tracking from a single fixed sensor, where it is shown in simulation to be robust against target confusion and other classically challenging data assignment problems [9]. The LocalMap extends that RBPF framework to track the motion of dynamic obstacles in an urban environment relative to a moving sensing platform, i.e. a ground robot, where nearby obstacles occupy much of the field of view and viewpoints of these obstacles change rapidly. The approach offers a more rigorous Bayesian solution to the joint estimation problem than maximum likelihood data assignments, and the solution is made practical through the RBPF's computationally efficient factorization of data assignment and obstacle tracking.

In the LocalMap, as with other particle filters derived for Monte Carlo data assignment, each randomly-drawn particle stores one possible data assignment history [28]. The particles are drawn according to a *proposal density* $q\left(A_k|Z_k\right)$, selected for its efficient sampling algorithms and its similarity to $p\left(A_k|Z_k\right)$. The particles drawn from this proposal density represent an approximation of the true density $p\left(A_k|Z_k\right)$ for the purposes of approximate inference:

$$p\left(A_k|Z_k\right) \approx \sum_i w_k^i \cdot \delta\big(A - A_k^i\big) \tag{3}$$

where w_k^i is the likelihood weight of the ith particle A_k^i at time index k:

$$w_k^i = \frac{p\left(A_k^i|Z_k\right)}{q\left(A_k^i|Z_k\right)} \tag{4}$$

and, because the data assignment problem is a discrete estimation problem, $\delta\left(\cdot\right)$ is the Kronecker delta function. Note the weights w_k^i must sum to unity to preserve the density's normalization:

$$\sum_i w_k^i = 1 \tag{5}$$

With equation 3 providing an approximate representation of $p\left(A_k|Z_k\right)$ and a closed-form parametric filter used to represent $p\left(O_k|Z_k, A_k\right)$, the full joint probability density from equation 2 is written approximately as:

$$p\left(A_k, O_k|Z_k\right) \approx \sum_i w_k^i \cdot \delta\big(A - A_k^i\big) \cdot p\left(O_k|Z_k, A_k^i\right) \tag{6}$$

where the obstacle densities $p\left(O_k|Z_k, A_k^i\right)$ are conditioned on the specific data assignment history A_k^i of a particular particle. In essence, each particle in equation 6 represents a complete hypothesis about the ego robot's urban environment. Each particle contains both a history of data assignment decisions as well as a parametric filter estimating states of obstacles whose existence are implied by the particle's measurement assignments. Since each particle may have a different data assignment history, particles may have different estimates of obstacle states and may even have different numbers of obstacles. Regardless, the factorization has simplified the urban perception problem dramatically. By exploiting the success of parametric filters such as the KF and SPF for target tracking under known measurement assignments, the LocalMap only needs to draw particles over the discrete data assignment portion of the joint density $p\left(A_k, O_k|Z_k\right)$.

Further simplifications to the dynamic urban perception problem can be made by realizing that the ego robot operating in real-time only requires an estimate of its surroundings at the present time. The problem is therefore simplified considerably by deleting measurement assignment and obstacle state histories from each particle after the information has been incorporated into the current estimate. This may be done by modifying the proposal density $q\left(A_k^i|Z_k\right)$, nominally a design choice 'similar to' the true distribution $p\left(A_k|Z_k\right)$, to delete old information. In particular, if the proposal density $q\left(A_k|Z_k\right)$ is chosen to factorize as follows:

$$q\left(A_k|Z_k\right) = q\left(a_k|Z_k, A_{k-1}\right) \cdot q\left(A_{k-1}|Z_{k-1}\right) \tag{7}$$

then the likelihood weight from equation 4 can be expressed recursively [28]:

$$w_k^i \propto \frac{p\left(z_k|Z_{k-1}, A_k^i\right) \cdot p\left(a_k^i|Z_{k-1}, A_{k-1}^i\right)}{q\left(a_k^i|Z_k, A_{k-1}^i\right)} \cdot w_{k-1}^i \tag{8}$$

where lowercase variables z_k and a_k indicate measurements and data assignments at a particular time index k. Equation 8 has been obtained, up to a normalization constant, by applying Bayes's rule to the data assignment density $p\left(A_k|Z_k\right)$. Notice that both terms in the numerator of equation 8 are familiar quantities: $p\left(z_k|Z_{k-1}, A_k^i\right)$ is the joint filter likelihood of the tracked obstacles after the kth data assignment decision in the ith particle, and $p\left(a_k^i|Z_{k-1}, A_{k-1}^i\right)$ is the predicted joint likelihood of the kth data assignment decision in the ith particle, before the kth measurement has actually been made.

One final step remains in deriving the particle filter for data assignment, and that is choosing the proposal density $q\left(a_k^i|Z_k, A_{k-1}^i\right)$ from which the random data assignment hypotheses for each particle are drawn. Because the density defines how the particles are sampled, it should be a density that can be sampled inexpensively. In particle filtering applications that estimate the state of a dynamic system, the proposal density is often chosen based only on the system's transition model and its process noise, thereby ignoring the measurement history [28]. This choice is suboptimal in the sense that it does not minimize the sample variance on the particles' weights w_k^i, which more rapidly drives all but a few particle weights to zero under repeated renormalizations. In contrast, the optimal proposal density $q_{\text{opt}}\left(\cdot\right)$ minimizing the sample variance on the particles' weights has been shown to be the true density [28]. For the LocalMap's data assignment problem, this density is:

$$q_{\text{opt}}\left(a_k^i|Z_k, A_{k-1}^i\right) = \alpha_k^i \cdot p\left(a_k^i|Z_k, A_{k-1}^i\right) \tag{9}$$

where α_k^i is a normalization constant explicitly included to make clear that the proposal density must sum to unity across the set of available data assignments in the ith particle. In the special case of data assignment, the optimal proposal density may be sampled directly and inexpensively [9]. To derive the corresponding sampling algorithm, the optimal density is first rewritten using Bayes's rule:

$$\begin{aligned} p\left(a_k^i|Z_k, A_{k-1}^i\right) &= \alpha_k^i \cdot p\left(z_k|a_k^i, Z_{k-1}, A_{k-1}^i\right) \\ &\cdot \; p\left(a_k^i|Z_{k-1}, A_{k-1}^i\right) \end{aligned} \tag{10}$$

where α_k^i is the normalizing constant, potentially different for each particle, that ensures all data assignment probabilities in the ith particle sum to unity. The first of the remaining two terms, $p\left(z_k|a_k^i, Z_{k-1}, A_{k-1}^i\right)$, is the likelihood of the measurement z_k, assuming it originated from a particular tracked obstacle in the ith particle. This term is nothing more than the filter likelihood for the tracked obstacle, which is often accurately assumed to be Gaussian in parametric filters [25]. The second term, $p\left(a_k^i|Z_{k-1}, A_{k-1}^i\right)$, is a transition model for the data assignments. This term represents any *a priori* assignment information that may be present before the measurement at time index k is actually made. In the present implementation of the LocalMap, this term is assigned a uniform probability across all data assignments. This choice is made to represent the fact that sensors are fused asynchronously in the LocalMap, and the measurements in general have no predictable ordering.

Equation 10 thus allows each particle to choose a random data assignment for the measurement z_k with assignment probabilities proportional to the measurement's likelihood of corresponding to each tracked obstacle in the particle. More precisely, the optimal proposal density from equation 10 is substituted into equation 8 to yield the final form of the weight update:

$$w_k^i = w_{k-1}^i \cdot \frac{1}{\alpha_k^i} \tag{11}$$

The remaining constant α_k^i is the normalizing constant for the optimal proposal density $q_{\text{opt}}(\cdot)$ from equations 9 and 10, left over after computing the likelihood ratio in equation 8 with the optimal proposal density. In the specific case where there is no prior information about the next measurement assignment, i.e. $p\left(n_k^i|Z_{k-1}, N_{k-1}^i\right)$ is uniform, the normalization constant is:

$$\alpha_k^i = \left[\frac{1}{M} \cdot \sum_{m=1}^{M} p\left(z_k|a_{m,k}^i\right)\right]^{-1} \tag{12}$$

where the sum is performed across all M obstacles tracked within the ith particle, and $a_{m,k}^i$ is the event that the measurement z_k taken at time k is assigned to the mth of M obstacles in the ith particle. Notice this final form of the LocalMap's weight update has the satisfying interpretation that particles whose obstacles better match the received sensor measurements have higher weights.

The sampling and reweighting calculations complete the data assignment particle filter defined by equation 3 and the RBPF defined by equation 6. A high-level description of the algorithmic steps taken to run the LocalMap RBPF for data assignment and obstacle tracking are given below:

1) Draw an initial set of particles A_0^i, $\forall i \in [1, N]$.
2) Predict all obstacles in each particle forward in time to the next measurement to yield a parametric representation of $p\left(O_k|Z_{k-1}, A_{k-1}^i\right)$.
3) For each particle, pick a data assignment for the measurement using the optimal proposal density in equation 10.
4) Update the parametric tracking filter to yield $p\left(O_k|Z_k, A_k^i\right)$ for the obstacle in each particle chosen to receive the measurement.
5) Update particle weights according to equation 11.
6) Resample particles to keep the filter well-conditioned, if necessary. Effective resampling strategies are discussed in Arulampalam *et al.* and Grisetti *et al.* [28], [30].
7) Go to step 2.

Occasionally, the LocalMap performs step 6, a particle resampling step, to alleviate degeneracy problems. These common particle filtering problems occur over time due to unlucky sampling choices that cause particle weights to degrade asymptotically to zero over repeated renormalizations [28]. In the LocalMap, such a problem might arise over time as the particles continue to sample random data assignment decisions without a means of correcting poor choices. If left unchecked, all particles in the LocalMap would eventually make poor data assignment decisions, leaving the filter with no particularly good hypothesis among its particles. To avoid this problem, the LocalMap's particles are occasionally resampled into a new set of particles with probability proportional to their weight. Particles with higher weights, which correspond to more likely interpretations of the urban environment, are more likely to be sampled into the new set of particles one or more times. Particles with lower weights, which correspond to unlikely hypotheses, may not even be sampled into the new set of particles at all. To decide when to resample, the LocalMap adopts the method proposed by Grisetti *et al.* for estimating the effective number of particles $\hat{N}_k$ that substantially influence the shape of the filtered density at time index k [30]:

$$\hat{N}_k = \frac{1}{\sum_i \left(w_k^i\right)^2} \tag{13}$$

For normalized particle weights drawn over N particles, equation 13 implies that $1 \leq \hat{N}_k \leq N$. Intuitively, $\hat{N}_k$ achieves its maximum when all particle weights are equal, which occurs when the particle filter assigns equal likelihood to each of its particles. Similarly, $\hat{N}_k$ achieves its minimum when all but one weight are zero, the undesirable situation in which the particle filter effectively places all likelihood on only one of its hypotheses. Grisetti *et al.* suggest thresholding $\hat{N}_k$ to determine whether resampling is necessary [30]. The LocalMap adopts this approach, resampling the particles when $\hat{N}_k \leq N/2$.

Thus far, it has been assumed that each particle is given a list of preordained obstacles O_0^i at $k = 0$ that are all present and visible from the moment the LocalMap is first started. In many practical situations this *a priori* information is not available, and instead the LocalMap must begin with a set of N empty particles: $O_0^i = \emptyset$, $\forall i \in [1, N]$. The creation of new obstacles is then handled within the particle filtered

data assignment framework. In particular, a birth likelihood $p\left(z_k|a^i_{m=0,k}\right)$ can be defined for each measurement z_k to represent the likelihood that the measurement originates from a newly-visible obstacle. The LocalMap's RBPF then considers this likelihood with those of any existing obstacles when choosing data assignments: the measurement is either assigned to an existing obstacle, or used to initialize a new one [23]. In contrast, the LocalMap does not remove obstacles from particles with a corresponding death likelihood, as each obstacle's existence depends on its state history as well as its assigned measurements. Obstacle death is therefore modeled in the LocalMap's parametric estimates of the obstacle density $p\left(O_k|Z_k, A_k\right)$, where, for practical reasons, it is implemented as a deterministic decision based on the obstacle's state and measurement history. The convenience of this formulation is that the LocalMap automatically determines the number of obstacles in the environment, and obstacles may pass in and out of sensor range without a growing computational burden.

B. The Continuous Tracking Problem

Section II-A describes the method used in the LocalMap to run an RBPF that samples a set of high likelihood data assignment hypotheses. This solution to the data assignment problem relies on the effectiveness of an inexpensive yet accurate parametric filter for tracking obstacles in the urban environment under known measurement assignments. The goal of this parametric filter is to describe obstacle geometry and motion as accurately as possible, making each obstacle distinct and distinguishable for measurement assignment. This reduces computational effort in the RBPF by decreasing the number of high likelihood data assignment hypotheses, and accordingly, the expected number of particles required to fully sample those hypotheses.

To further reduce the computational burden of the problem, all obstacles are assumed to be conditionally independent given sensor measurements and their assignments. This allows the continuous tracking problem to be factorized into a set of single obstacle tracking problems:

$$p\left(O_k|Z_k, A^i_k\right) = \prod_m p\left(O_{m,k}|Z_k, A^i_k\right) \tag{14}$$

where $p\left(O_{m,k}|Z_k, A^i_k\right)$ is the probability density of the mth obstacle in the ith particle. By assuming conditional independence among the obstacles, each particle in the LocalMap uses a bank of small parametric filters, one for each obstacle tracked. This factorization offers substantial computational savings, since parametric filters often rely on expensive matrix operations cubic in the size of the filter's state vector. The alternative, a single large parametric filter simultaneously estimating the joint state of all obstacles in a particle, would be prohibitively expensive. The parametric filter described in this section therefore sets out to track a single obstacle, potentially moving, under known measurement assignments.

Since the LocalMap is designed to solve the dynamic urban perception problem, all obstacles are assumed to have size, shape, and motion similar to full-size motor vehicles. This assumption is made to constrain the LocalMap's attention to vehicular traffic, though it does not preclude an expanded list of obstacle classes including pedestrians, bicyclists, tractor-trailers, and other dynamic objects commonly present in urban environments. The benefit of adopting the former constraint is that it avoids an ancillary target classification problem, as each obstacle is tracked with an instance of the same parametric filter.

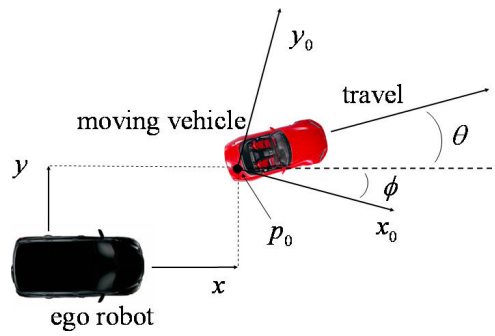

Fig. 1. Coordinate frames used for tracking a single obstacle (a moving vehicle) under known measurement assignments. In the LocalMap, an obstacle's position is defined by the x and y location of a reference point p_0 fixed to the obstacle.

The most common parametric filter used for tracking a single full-size moving vehicle models the vehicle as a rectangle [2], [15], [16], [22], [23]. Such a filter typically estimates the vehicle's length and width in addition to motion parameters such as position and velocity. Problems with this approach stem from two sources: the fact that vehicles are not rectangles, and the fact that no sensor measures the entire vehicle as if it were. Unfortunately, these two problems have competing solutions: the rectangle model is not rich enough to describe a moving vehicle, but it is too complex to permit simple Bayesian updates. Many existing approaches strike a compromise by accepting the former problem and applying an *ad hoc* feature extraction or matching algorithm to raw sensor data to address the latter [2], [12], [15], [16]. However, as MacLachlan and Mertz point out, these feature extraction techniques yield erroneous motion estimates due to the substantial change in the sensed shape of a vehicle as it moves around the sensor's field of view [17].

The LocalMap tracking algorithm avoids these problems with a filter that operates directly on raw sensor data. The filter contains five state variables describing the position, motion, and shape of the moving vehicle in a coordinate frame fixed to the ego robot. Figure 1 shows how these state variables are used to define the moving vehicle within the parametric filter. Internally, the filter's tracked vehicle is stored as two pieces of data: a parameterized probability density $p\left(O_{m,k}|Z_k, A^i_k\right)$ over the vehicle's state $O_{m,k}$, and a cloud of sensed points describing the vehicle's shape. Because it is sensed rather than parameterized, this point cloud representation stores a far more accurate description of the tracked vehicle's geometry than more common rectangular models. A more subtle observation is the fact that integrating the point cloud representation is no more expensive than integrating parameterized and highly simplified vehicle geometries, because the tracked vehicle is a rigid body. As a result, the position and motion of all points

in the cloud are related through an unchanging set of affine transformations. If the relative positions of these points are stored in an obstacle-fixed coordinate frame, the motion of one such reference point in the ego robot frame is sufficient to reconstruct the motion of all points. A further key observation is that direct measurements of this one target-fixed reference point, labeled p_0 in Figure 1, are not necessary to accurately estimate the motion of the target vehicle. Instead, the position of the fixed reference point is included in the set of states to be estimated. The resulting tracking filter produces consistent motion estimates by utilizing sensor information within the Bayesian framework to simultaneously estimate the location and motion of an arbitrary fixed reference point on the vehicle being tracked. Because the filter does not rely on measuring a *specific* fixed reference point, such as the tracked vehicle's center of mass, it avoids estimation artifacts that arise from *ad hoc* attempts to locate these specific points in raw sensor data.

The proposed filter parameterizes a tracked vehicle's position, motion, and shape with five states:

$$o_{m,k} = [\, x_k \quad y_k \quad s_k \quad \theta_k \quad \phi_k \,]^T \tag{15}$$

where, from now on, $o_{m,k}$ will be used instead of $O_{m,k}$ to reflect the desire to estimate the state of the tracked vehicle only at the current time step k. From Figure 1, the position variables x and y describe the location of the tracked vehicle's arbitrary fixed point relative to the ego robot. The motion of the tracked vehicle's fixed point is parameterized by the velocity variables s and θ, which store the point's absolute ground speed and heading relative to the ego robot. The shape of the tracked vehicle is parameterized by ϕ, which describes the rigid body rotation angle between the ego robot's coordinate frame and the obstacle-fixed frame in which the tracked vehicle's point cloud is stored. Intuitively, ϕ accumulates the total change in the angle at which the vehicle is observed while it is being tracked. Note that if the tracked vehicle's point cloud is stored relative to the fixed point, the combined rotation ϕ and translation $[\, x \quad y \,]^T$ suffice to locate any point in the point cloud with respect to the ego robot. Similarly, the transformation may be combined with the motion parameters s and θ to compute the motion of any point in the tracked vehicle's point cloud.

The time evolution of $o_{m,k}$ in the (potentially moving) coordinate frame fixed to the ego robot is modeled by the following system of continuous-time nonlinear differential equations:

$$\begin{aligned}
\dot{x} &= s \cdot \cos(\theta) - v_x + y \cdot \omega_z + e_x && (16)\\
\dot{y} &= s \cdot \sin(\theta) - v_y - x \cdot \omega_z + e_y && (17)\\
\dot{s} &= e_s && (18)\\
\dot{\theta} &= -\omega_z + e_\theta && (19)\\
\dot{\phi} &= -\omega_z + e_\phi && (20)
\end{aligned}$$

where v_x and v_y are components of the ego robot's velocity, ω_z is the ego robot's rate of rotation, and e_x, e_y, e_s, e_θ, and e_ϕ are zero mean, mutually uncorrelated, Gaussian, white random variables acting as *process noise*. Intuitively, these random variables account for unmodeled maneuvers executed by the tracked vehicle. Note that although equation 18 assumes the tracked vehicle moves on average in a straight line at constant speed, any parameterized dynamics model may be used. This flexibility is a major benefit of the LocalMap's point cloud representation: the dynamics model merely describes the time evolution of the rigid body transformation between the coordinate frame fixed to the ego robot and the one fixed to the tracked vehicle.

In addition to the aforementioned process noise accounting for vehicle maneuvers, it is noted that v_x, v_y, and ω_z are measured by noisy sensors. As a result, the following substitutions are made:

$$\begin{aligned}
v_x &= \hat{v}_x + e_{v_x} && (21)\\
v_y &= \hat{v}_y + e_{v_y} && (22)\\
\omega_z &= \hat{\omega}_z + e_{\omega_z} && (23)
\end{aligned}$$

where $\hat{v}_x$, $\hat{v}_y$, and $\hat{\omega}_z$ are measured from odometry sensors on the ego robot, and e_{v_x}, e_{v_y}, and e_{ω_z} are zero mean, mutually uncorrelated, Gaussian, white random variables acting as additional process noise. Regrettably, these noisy estimates of ego robot motion result in correlations between all obstacles tracked in the LocalMap, so that equation 14 is only an approximate factorization. In SLAM literature, where obstacles are static and often modeled with no process noise, such correlations are large and central to the localization problem [5]. In a dynamic environment, the opposite is true: obstacles are modeled with significant uncorrelated process noise to capture uncertainty in maneuvers such as acceleration and turning. This process noise effectively swamps correlations between obstacles, as uncertainty in their maneuvers is far larger than uncertainty in commonly-available automotive odometry sensors. As a result, the factorization in equation 14 is taken as a valid approximation.

To facilitate the use of a computationally inexpensive parametric filter, $p\left(o_{m,k} | Z_k, A_k^i\right)$ is assumed to be Gaussian. In light of the weak nonlinearities in equations 16 - 20, the time evolution of this Gaussian is computed in the LocalMap using the *prediction step* of the Extended Kalman Filter (EKF). The prediction step is implemented using a fourth order Runge-Kutta numerical integration to convert equations 16 - 20 to a nonlinear discrete time difference equation of the form:

$$o_{m,k+1} = f\left(o_{m,k}, v_{m,k}\right) \tag{24}$$

where $v_{m,k}$ is a vector of zero mean, mutually uncorrelated, Gaussian, white random variables derived from e_x, e_y, e_s, e_θ, e_ϕ, e_{v_x}, e_{v_y}, and e_{ω_z}. Traditional EKF equations are then used to compute the time evolution $p\left(o_{m,k+1} | Z_k, A_k^i, \hat{v}_{x,k}, \hat{v}_{y,k}, \hat{\omega}_{z,k}\right)$ conditioned on past measurements, assignments, and the measured motion of the ego robot [25]. Although conditioning on measurements of ego robot motion is henceforth suppressed for brevity, it is understood to be present in the time evolution of all obstacles.

When a new sensor measurement z_{k+1} is assigned to the tracked vehicle, the vehicle's posterior probability density can be updated to $p\left(o_{m,k+1} | Z_{k+1}, A_{k+1}^i\right)$ to reflect the new measurement. The exact form of the update depends greatly on the information contained in the measurement, and therefore on

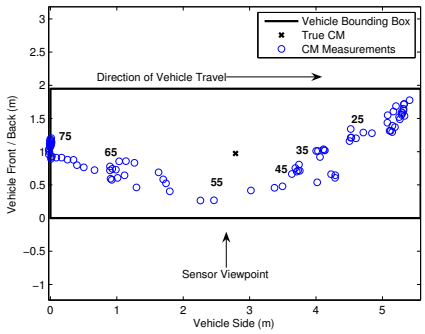

Fig. 2. Locus of point cloud centers of mass observed as a moving vehicle passes in front of a stationary laser rangefinder. The apparent drift in the observed center of mass results from dramatic changes in the sensed shape of the moving vehicle as it passes in front of the laser rangefinder. This sensitivity, appearing in even mildly dynamic environments, causes instability in feature extraction algorithms such as center of mass, edge, and corner detectors applied to laser rangefinder data.

the type of sensor generating the measurement. Three popular automotive sensing modalities are explored for the LocalMap: laser rangefinder, radar, and optical camera. Measurement updates for these sensing modalities are discussed in Sections II-B1 and II-B2.

1) Laser Rangefinder: The first type of sensor fused in the LocalMap is the laser rangefinder, which measures a point cloud of returns generated from patches in the environment reflecting the sensor's emitted energy. Though laser rangefinders offer centimeter-level ranging and sub-degree bearing accuracy, a practical problem arises due to the fact that they measure individual points rather than entire objects. Existing approaches solve the problem by extracting features from the point cloud, such as lines, corners, rectangles, or the center of mass, and then using these features as measurements to update the parameterized probability density of the tracked vehicle [2], [14]–[18]. Unfortunately, these feature extraction algorithms are unstable in even mildly dynamic environments, where the shape and motion of tracked vehicles and pitch and roll of the ego robot cause rapid changes in object shape. Unmodeled instability in these features frequently yields erroneous state estimates, particularly in the motion of the tracked vehicle. Figure 2 illustrates the problem by plotting the locus of point cloud centers of mass observed as a moving vehicle passes in front of a stationary laser rangefinder. Here, the vehicle's motion causes the observed center of mass to drift with respect to the true center of mass. On average the observed center of mass lies more than 2 m from the true center of mass, and all measurements are at least 60 cm from the truth. Numeric indices of center of mass measurements in Figure 2 also show the motion of the observed measurements is largely opposite to the vehicle's direction of travel.

Two alternatives exist to alleviate estimation problems stemming from instability in sensed features. The first alternative is to model the instability, by accounting for the extra uncertainty in the measurement likelihood $p\left(z_{k+1}|o_{m,k+1}, A^i_{k+1}\right)$. While this solution may yield consistent state estimates, it is unsatisfactory in the sense that the resulting estimation errors are larger to account for measurement instability. The second alternative, adopted for laser rangefinder data fused in the LocalMap, is to use an alternate set of measurements that are more reliably stable. First, the laser points are grouped into distinct objects via a clustering algorithm. The exact clustering algorithm used is not important, though the conservative algorithms presented in Miller and Campbell [23] or Miller *et al.* [31] yield stable performance in practice. After the laser points are clustered into distinct objects, three measurements are extracted: the smallest and largest bearings b_{min} and b_{max} from each cluster, and the range r_{min} to the closest point in each cluster. A measurement vector z_{k+1} consists of these three measurements extracted from a single cluster:

$$z_{k+1} = \left[\, b_{\text{min}} \quad b_{\text{max}} \quad r_{\text{min}} \,\right]^T \tag{25}$$

and depending on the number of clusters present, multiple measurement vectors may be extracted from a single frame of laser returns.

The most important aspect of the bearing-bearing-range measurement described in equation 25 is its stability: the values of b_{min}, b_{max}, and r_{min} change slowly as the tracked vehicle's point cloud undergoes small translations and rotations. This property is not shared by other extracted features, such as the point cloud's center of mass, which may suffer large discontinuities in the face of small transformations. Such strong nonlinearities make those features unsuitable for linear estimation techniques.

In contrast, the weak nonlinearities in the bearing-bearing-range measurement make it ideal for use in linear estimation techniques, which linearize the relationship between the measurement and the state vector. For the LocalMap, this relationship is made explicit with the following auxiliary variables:

$$\begin{aligned} \beta_{\text{min}} &= \min_{p \in P_{m,k+1}} \angle\left[T_s\left(x, y, \phi\right) \cdot \left(p - p_s\right)\right] \\ \beta_{\text{max}} &= \max_{p \in P_{m,k+1}} \angle\left[T_s\left(x, y, \phi\right) \cdot \left(p - p_s\right)\right] \\ \rho_{\text{min}} &= \min_{p \in P_{m,k+1}} \left\|T_s\left(x, y, \phi\right) \cdot \left(p - p_s\right)\right\| \end{aligned} \tag{26}$$

where β_{min}, β_{max}, and ρ_{min} constitute the state-generated measurement, $P_{m,k+1}$ is the set of points corresponding to the mth tracked vehicle, $T_s\left(\cdot\right)$ is the transformation matrix that projects the point cloud $P_{m,k+1}$ into the coordinate frame of the laser rangefinder, p_s is the location of the laser rangefinder, and the operators $\angle\left(\cdot\right)$ and $\|\cdot\|$ return the bearing and magnitude of their vector arguments, respectively. With these auxiliary variables defined, the bearing-bearing-range measurement in equation 25 relates to the mth tracked vehicle's state through the nonlinear *measurement function* $h_L\left(\cdot\right)$:

$$h_L\left(o_{m,k+1}, P_{m,k+1}\right) = \begin{bmatrix} \beta_{\text{min}} \\ \beta_{\text{max}} \\ \rho_{\text{min}} \end{bmatrix} \tag{27}$$

With the measurement function defined, the measurement likelihood $p\left(z_{k+1}|o_{m,k+1}, A^i_{k+1}\right)$ is modeled as Gaussian, corrupted by additive zero mean Gaussian white noise w_{k+1}:

$$z_{k+1} = h_L\left(o_{m,k+1}, P_{m,k+1}\right) + w_{k+1} \tag{28}$$

where the Gaussian measurement noise w_{k+1} is modeled as additive to reflect the fact that the laser rangefinder measures bearings and ranges directly. With $p\left(z_{k+1}|o_{m,k+1}, A^i_{k+1}\right)$ modeled as Gaussian, linear measurement updating techniques can be used to update $p\left(o_{m,k+1}|Z_k, A^i_{k+1}\right)$ to $p\left(o_{m,k+1}|Z_{k+1}, A^i_{k+1}\right)$ to reflect the new measurement. In particular, the LocalMap utilizes the Sigma Point Transform in the update step of the SPF, because numerical differentiation of $h_L\left(o_{m,k+1}, P_{m,k+1}\right)$ is more convenient than explicit differentiation. This measurement is an ideal complement to the point cloud representation of the tracked vehicle, as neither rely on knowledge of a particular fixed point on the tracked vehicle. Instead, information about the tracked vehicle's motion and evolving rigid body transformation are gathered indirectly from robust measurements via fusion in a Bayesian tracking filter.

2) Radar And Optical Camera: The second and third types of sensors fused in the LocalMap are radar and optical cameras. Like the laser rangefinder, measurements from these sensors are fused in the LocalMap at the object level. In other words, the LocalMap relies on external processing to group raw sensor data into measurements of distinct objects. In the case of radar and optical camera data, this processing is commercially available; radar and camera systems built for collision detection often process raw data into objects. For the radar, the LocalMap utilizes measurements of the tracked vehicle's bearing, range, and range rate in the sensor's coordinate frame:

$$z_{k+1} = \left[\, b_s \quad r_s \quad \dot{r}_s \,\right]^T \tag{29}$$

and as with equation 25, multiple measurements may be available if more than one vehicle is present.

Much like the laser rangefinder, the radar measurement suffers from ambiguity. The radar's range and bearing components do not measure a particular fixed point on the tracked vehicle, and in fact the measured point may change over time as reflective portions of the vehicle become visible or occluded. The difficulty is further compounded by the fact that the radar's algorithm for generating bearing and range measurements is often unknown, as many off-the-shelf radar units do not document their internal measurement processing algorithms. As a result, there is no way to relate the radar's measurements to the tracked vehicle's point cloud to eliminate the measurement ambiguity completely. Instead, the tracked vehicle's point cloud is used to generate an approximate measurement function $h_R\left(o_{m,k+1}, P_{m,k+1}\right)$ for the radar:

$$h_R\left(\cdot\right) = \begin{bmatrix} \frac{1}{2}\left(\beta_{\text{min}} + \beta_{\text{max}}\right) \\ \rho_{\text{min}} \\ \left\{v_v\left(s,\theta\right) - v_s\left(\hat{v}_x, \hat{v}_y, \hat{\omega}_z\right)\right\} \cdot e_r\left(x,y\right) \end{bmatrix} \tag{30}$$

where $v_v\left(\cdot\right)$ is the velocity of the tracked vehicle, $v_s\left(\cdot\right)$ is the velocity of the sensor on the ego vehicle, and $e_r\left(\cdot\right)$ is the unit vector from the radar to the tracked vehicle. As with the laser rangefinder, the measurement likelihood $p\left(z_{k+1}|o_{m,k+1}, A^i_{k+1}\right)$ is modeled as Gaussian, corrupted by additive zero mean Gaussian white noise w_{k+1}:

$$z_{k+1} \approx h_R\left(o_{m,k+1}, P_{m,k+1}\right) + w_{k+1} \tag{31}$$

where in approximation, the radar measures the center bearing, the closest range, and the range rate of the tracked vehicle's arbitrary fixed point. The ambiguity of the radar measurement is then addressed in the covariance matrix of the measurement noise w_{k+1}, where bearing and range measurement noise standard deviations are set large enough to account for the fact that the measurements can correspond to any point on the tracked vehicle. As with the laser rangefinder, the Sigma Point Transform of the SPF update is then used to generate $p\left(o_{m,k+1}|Z_{k+1}, A^i_{k+1}\right)$ from the radar measurement.

Optical camera measurements are incorporated into $p\left(o_{m,k+1}|Z_k, A^i_{k+1}\right)$ similar to radar measurements. For the optical camera, the LocalMap utilizes measurements of the tracked vehicle's position, range rate, and width in the sensor's coordinate frame:

$$z_{k+1} = \left[\, x_s \quad y_s \quad \dot{r}_s \quad w_s \,\right]^T \tag{32}$$

where $\left(x_s, y_s\right)$ is the tracked vehicle's location, $\dot{r}_s$ is the vehicle's range rate, and w_s is the width of the tracked vehicle in the image plane.

Unlike the laser rangefinder and the radar, the optical camera measurement is not necessarily ambiguous. If edge or symmetry kernels are used to find the tracked vehicle's centerline as a position measurement, then the measurement is not ambiguous. In fact, the sensor-driven measurement of the centerline bearing b_{avg} is related to the minimum and maximum bearing measurements used in equation 25:

$$b_{\text{avg}} = \frac{1}{2}\left(b_{\text{min}} + b_{\text{max}}\right) \tag{33}$$

if the edge or symmetry kernels discover the same boundary as the laser rangefinder clustering algorithm. In contrast, the model-driven centerline bearing β_{avg} from the point cloud representation is:

$$\beta_{\text{avg}} = \frac{1}{2}\left(\beta_{\text{min}} + \beta_{\text{max}}\right) \tag{34}$$

Using β_{avg} and the other stable measurements, the optical camera's measurement function $h_C\left(o_{m,k+1}, P_{m,k+1}\right)$ is:

$$h_C\left(\cdot\right) = \begin{bmatrix} \rho_{\text{min}} \cdot \cos\left(\beta_{\text{avg}}\right) \\ \rho_{\text{min}} \cdot \sin\left(\beta_{\text{avg}}\right) \\ \left\{v_v\left(s,\theta\right) - v_s\left(\hat{v}_x, \hat{v}_y, \hat{\omega}_z\right)\right\} \cdot e_r\left(x,y\right) \\ 2 \cdot \rho_{\text{min}} \cdot \tan\left(\frac{1}{2}\left\{\beta_{\text{max}} - \beta_{\text{min}}\right\}\right) \end{bmatrix} \tag{35}$$

Using the measurement function, the camera measurement likelihood $p\left(z_{k+1}|o_{m,k+1}, A^i_{k+1}\right)$ is also modeled as Gaussian, corrupted by additive zero mean Gaussian white noise w_{k+1}:

$$z_{k+1} = h_C\left(o_{m,k+1}, P_{m,k+1}\right) + w_{k+1} \tag{36}$$

and the Sigma Point Transform of the SPF update is used to generate $p\left(o_{m,k+1}|Z_{k+1}, A^i_{k+1}\right)$ from the camera measurement.

III. Experimental Performance

The LocalMap tracking algorithm has been implemented in real-time on Cornell University's 'Skynet,' an autonomous 2007 Chevrolet Tahoe shown in Figure 3. Skynet is equipped with 7 laser rangefinders: 3 Ibeo ALASCA XTs in the front

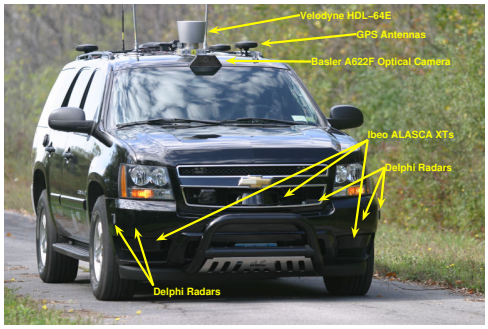

Fig. 3. The LocalMap is implemented in real-time on Cornell University's 'Skynet,' an autonomous 2007 Chevrolet Tahoe that completed the 2007 DARPA Urban Challenge.

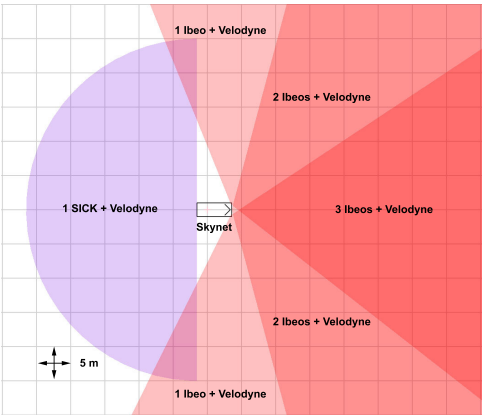

Fig. 4. Sensor placement and coverage diagram for Skynet's laser rangefinders. Skynet sits in the center of the diagram, facing right. Redundant laser rangefinders cover the front of Skynet's field of view, where it encounters the most dangerous moving obstacles.

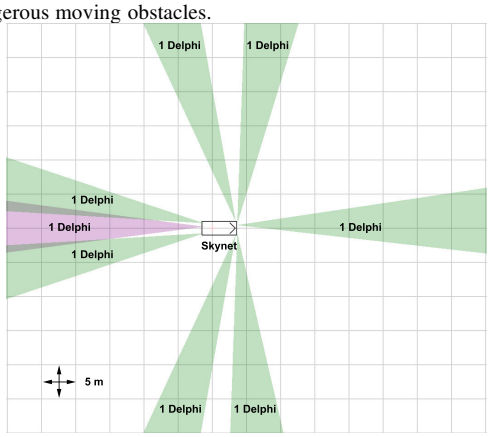

Fig. 5. Sensor placement and coverage diagram for Skynet's radars. Skynet sits in the center of the diagram, facing right. Radars are placed to detect oncoming cars in opposing lanes, from the left and right in merging situations, and from the rear in passing situations.

bumper, 1 SICK LMS 220 in the back bumper, 2 SICK LMS 291s in the rear driver and passenger doors, and a Velodyne HDL-64E on the roof. The placement and coverage of these laser rangefinders is shown in Figure 4. Skynet is also equipped with 8 Delphi FLR millimeter-wave radar units: 5 in the front bumper, and 3 in the back bumper. The placement and coverage of these radars is shown in Figure 5. Finally, Skynet is equipped with 2 optical cameras: a forward-facing Basler A622F, and a backward-facing Unibrain Fire-i 520b, both mounted on the roof and running MobilEye SeeQ vehicle tracking software. All sensors are accurately time-stamped via synchronized microcontroller interfaces, and each is available for obstacle detection and tracking in the LocalMap [31]. Skynet is also equipped with a position, velocity, and attitude estimator that fuses GPS with onboard vehicle odometry and inertial navigation to generate accurate and robust localization and differential vehicle motion estimates [32].

For evaluation purposes, the LocalMap has been implemented in C++ and connected to Skynet's time-stamped data logs. In this practical implementation, the LocalMap is initialized with no prior information. A birth likelihood is then used to discover new obstacles in the RBPF framework as per Section II-A, with sensor-dependent uniform likelihoods used to represent the likelihood of observing a new obstacle from each of Skynet's sensors. Similar to the birth model, each measurement is also evaluated against a clutter model created for each of Skynet's sensors. The corresponding clutter likelihood captures common sensor errors with uniform, Gaussian, and multi-modal Gaussian densities that account for multiple reflections, signal multipath, and false positives in each sensor. Deletion of old obstacles is similarly tied to the Bayesian framework, but implemented with deterministic thresholds to guard against the danger of randomly deleting a threatening obstacle. Each obstacle's existence is modeled as a probability that decays exponentially to zero with a 95% time constant of 3 seconds. Each measurement assigned to an obstacle provides evidence, in a Bayesian sense, for that obstacle's existence, and an occupancy grid created from Skynet's Velodyne provides evidence against false positives. Obstacles are removed from the LocalMap when their existence probability drops below 5%.

Specific practical accommodations are also made to adapt the point cloud representation across the LocalMap's three sensing modalities. Since neither the Delphi radars nor the MobilEye SeeQ software generate point clouds, obstacles that have not yet been assigned laser rangefinder measurements are tracked with parametric filters only, i.e. with no point clouds. In addition, new point clouds assigned to an obstacle always overwrite existing point clouds, but only after the measurement update is performed in the obstacle's parametric filter. In other words, no attempt is made to merge multiple point clouds over time; only the obstacle's rigid body transform and motion parameters undergo Bayesian estimation. Two reasons support this design choice: the accuracy and scanning rates of laser rangefinders make it unnecessary to combine multiple point clouds, and the computational burden required to store and process increasingly large point clouds would quickly overwhelm any real-time implementation.

The LocalMap is tested in a series of three experiments. The first two experiments evaluate the tracking capabilities of the LocalMap in common but difficult maneuvers: a perpendicular intersection encounter similar to the circumstances of Figure 2, and a parallel head-on encounter. In these experiments, the LocalMap is evaluated against truth data obtained from pose estimators on board the ego vehicle and the target

Fig. 6. In the first experiment, the LocalMap tracks a moving target vehicle as it crosses the ego vehicle's path at an intersection.

vehicle being tracked. These estimators fuse traditional GPS and differential corrections with inertial navigation sensors on board both vehicles to produce position, velocity, and attitude estimates for both the ego and target vehicle with sub-meter position accuracy, cm / sec. velocity accuracy, and sub-degree attitude accuracy [32]. The LocalMap simultaneously tracks other obstacles in the environment aside from the target vehicle, though ground truth for these obstacles is unavailable and therefore not evaluated. These experiments are performed at the Seneca Army Depot in Romulus, New York, which contains a variety of features including paved and unpaved roads, some painted road lines, potholes, railroad tracks, considerable short and tall vegetation, and storage buildings.

The third experiment evaluates the LocalMap's consistency in tracking multiple obstacles in a densely-populated, highly dynamic environment. In this experiment, variants of the LocalMap algorithm with different numbers of particles are run on the same segment of logged data, a 19 minute excerpt of a DARPA Urban Challenge qualifying round at George Air Force Base in Victorville, California. The data contains Skynet's sensor measurements of two concentric loops of heavy traffic traveling in opposite directions, recorded as Skynet merged into and out of this traffic multiple times. No truth data is available for the qualifying round, so consistency among the LocalMap variants is used to evaluate their performance.

A. Experiment 1: Perpendicular Intersection Encounter

The first experiment evaluates the LocalMap's ability to track a moving target vehicle as it crosses the ego vehicle's path at an intersection. In this experiment, the ego vehicle (Skynet) remains stationary at the intersection, while the target vehicle drives past at approximately 15 mph. The LocalMap is run on the collected sensor data with 50 particles, and each particle is allowed to track as many obstacles as it sees fit. Data is collected for a total of 22 perpendicular intersection encounters: 11 with the target vehicle approaching from the right, and 11 with the target vehicle approaching from the left. Figure 6 shows the experimental setup.

When evaluating the LocalMap's tracking performance on this data, comparisons are made to the LocalMap's maximum *a posteriori* (MAP) estimate of the environment, which is the particle with the largest weight. Within the most likely particle, evaluations are made with the tracked obstacle that most closely matches the truth data. This obstacle is chosen according to a minimum Mahalanobis distance criterion weighing minimum range, and minimum and maximum bearing. Weights are chosen such that a simultaneous range error of 5 m and bearing errors of 5° yield unit distance. Any LocalMap iterations in which no tracked obstacle has distance ≤ 7.8147, which corresponds to a 95% confidence bound on a χ^2 random variable with 3 degrees of freedom, are considered missed detections and discarded. A total of 11 such missed detections occurred over the 28100 LocalMap iterations considered in this experiment.

LocalMap tracking statistics averaged over the 11 from-the-right encounters are shown in Figures 7, 8, and 9. In each Figure, statistics are parameterized by the true bearing of the target vehicle's front bumper to align the 11 trials despite minor variations in maneuver duration. The Figures read left to right: the target vehicle approaches the intersection at negative bearings, crosses the ego vehicle's path at bearing zero, and departs the intersection at positive bearings. The target vehicle is closest to the ego vehicle near bearing zero. Figure 7 plots average LocalMap errors in range to closest point on the moving target vehicle observed in the 11 intersection encounters. Predictably, ranging errors are smallest in the middle of each encounter, when the target vehicle is within 40 m (between -75° and 77°) of the ego vehicle and visible both by side-facing radars and laser rangefinders. Ranging errors increase slightly between approximately 0° and 40°, where the target vehicle is observed by Skynet's forward-facing radar. This radar occasionally provides erroneous information during perpendicular intersection encounters, as the target vehicle only travels perpendicular to its radial direction. These significantly incorrect measurements yield larger errors temporarily in one or two encounters, resulting in larger sample standard deviations. Errors are largest at the beginning of the maneuver, when the target vehicle is first observed between 60 and 80 m from the ego vehicle. Statistically, LocalMap ranging errors are within 20 cm of zero at the 5% significance level in 251 of the 284 bearings considered.

Figure 8 plots average LocalMap target vehicle ground speed estimation errors over the 11 from-the-right perpendicular intersection encounters. Speed estimation errors remain low from first acquisition through its approach, showing the LocalMap's ability to combine accurate radar speeds with accurate laser ranges to produce an obstacle estimate accurate in both speed and position. Accurate speeds are maintained even as the target vehicle crosses in front of the ego vehicle and out of view of the radars. Small speed errors are incurred temporarily as the target vehicle departs the intersection, where Skynet's left Ibeo laser rangefinder is the primary source of laser data. Errors incurred at these bearings appear to be the result of an angular miscalibration in the offending laser rangefinder, resulting in disagreement between Skynet's left and center laser rangefinders. Since the disagreement is in sensor yaw, it creates an unmodeled bias in the evolution of the target vehicle's point cloud while minimally affecting Skynet's estimates of range. The errors are reduced to normal at bearings near 80°, when it is too far away for the laser

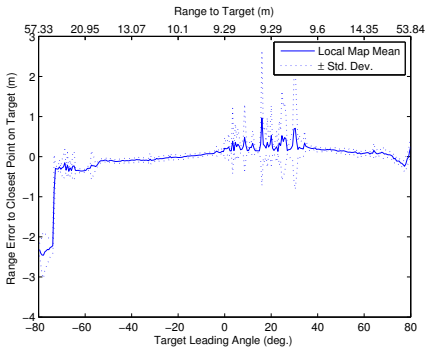

Fig. 7. LocalMap range tracking errors to the closest point on a moving target vehicle over 11 perpendicular intersection encounters with the target approaching from the right at 15 mph. Error statistics are calculated across encounters according to the true bearing of the target's leading edge. Of the 284 average range errors considered, 251 are within 20 cm of zero at the 5% significance level.

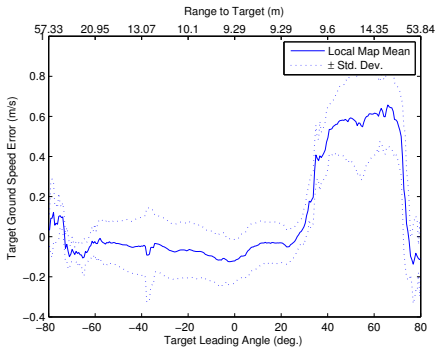

Fig. 8. LocalMap ground speed tracking errors over 11 perpendicular intersection encounters with a moving target vehicle approaching from the right at 15 mph.

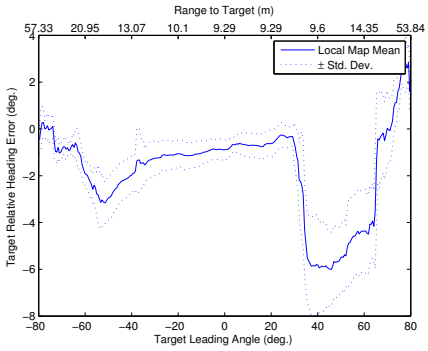

Fig. 9. LocalMap relative heading tracking errors over 11 perpendicular intersection encounters with a moving target vehicle approaching from the right at 15 mph. The LocalMap's point cloud representation is able to estimate relative heading correctly despite significant changes in target vehicle viewpoint.

rangefinders to produce usable clusters. Statistically, LocalMap ground speed estimation errors are indistinguishable from zero at the 5% significance level in 142 of 284 bearings considered, and 211 of 254 are within 0.1 m/sec. of zero at the same significance level.

Figure 9 plots average LocalMap target vehicle relative heading estimation errors over the 11 from-the-right perpendicular intersection encounters. Like the ground speed estimation errors, the target vehicle relative heading estimation

Fig. 10. In the second experiment, the LocalMap tracks a moving target vehicle as it approaches the moving ego vehicle from the opposite direction. Both vehicles travel at approximately 15 mph for these parallel head-on encounters.

errors remain relatively small from first acquisition through the intersection. It is noted, however, that these errors appear slightly biased, perhaps reflecting a minor angular miscalibration in the ego vehicle's right-facing laser. Estimated relative headings also suffer larger errors at target vehicle bearings between 40° and 80° due to Skynet's first left-facing radar. Statistically the overall heading errors are quite small: the LocalMap produces heading errors less than 2° at the 5% significance level in 210 of the 284 bearings considered.

B. Experiment 2: Parallel Head-On Encounter

The second experiment evaluates the LocalMap's ability to track a moving target vehicle from a moving ego vehicle as they approach each other from opposite directions on parallel tracks. In this experiment, the ego vehicle (Skynet) and the target vehicle approach each other in opposite lanes on a straight road. Both vehicles travel at approximately 15 mph during the experiment, for a combined closing speed of approximately 30 mph. The LocalMap is again run with 50 particles, and each particle is allowed to track as many obstacles as it deems appropriate. Data is collected for 11 such parallel head-on encounters, all conducted on the same road from the same vehicle starting positions. Figure 10 shows the experimental setup.

Like the first experiment, comparisons in the head-on encounters are made to the obstacles tracked in the LocalMap's MAP estimate of the environment. The same Mahalanobis distance criterion is used to choose the obstacle within this MAP estimate that most closely matches the truth data, and the same distance threshold is applied to discard frames in which the LocalMap was not tracking the target vehicle. The LocalMap experienced 231 of these missed detection frames among the 7264 considered in this experiment. All but one of these missed detection frames occurred when the target vehicle was more than 95 m from the ego vehicle.

LocalMap tracking statistics averaged over the 11 parallel head-on encounters are shown in Figures 11, 12, and 13. In this experiment, tracking data from the trials is aligned by the true range to the closest point on the target vehicle. Each Figure therefore reads right to left; the target vehicle enters detection range at approximately 120 m and approaches the ego vehicle in the oncoming lane until both pass each other. Evaluation

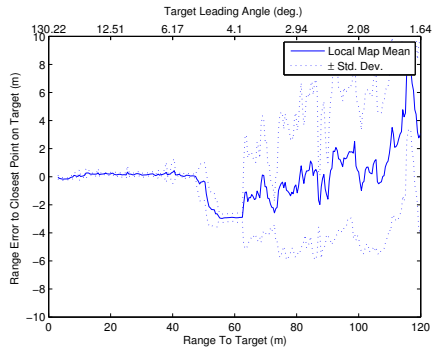

Fig. 11. LocalMap range tracking errors to the closest point on a moving target vehicle over 11 parallel head-on encounters at 30 mph closing speeds. Error statistics are calculated across encounters according to the true range to the target vehicle. In the 284 true ranges to the target vehicle considered, 153 are statistically equal to zero, and 233 are within 20 cm of zero at the 5% significance level.

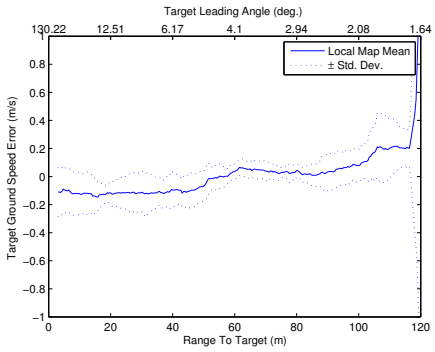

Fig. 12. LocalMap ground speed tracking errors over 11 parallel head-on encounters at 30 mph closing speeds.

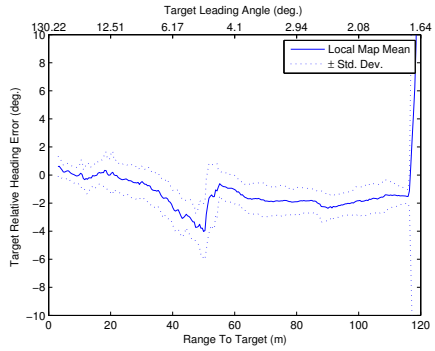

Fig. 13. LocalMap relative heading tracking errors over 11 parallel head-on encounters at 30 mph closing speeds.

of the maneuver ends when the range to the closest point on the target vehicle is at a minimum, approximately 3 m. Figure 11 plots average LocalMap errors in range to closest point on the moving target vehicle observed in the 11 parallel head-on encounters. Range errors are largest when the maneuver begins, when only a single radar observes the target vehicle. Range errors drop to near zero at ranges of approximately 50 m, when the target vehicle is first observed by laser rangefinders. The LocalMap also becomes more accurate and consistent with the additional sensor data at these ranges, as shown by its small standard deviations across the trials. Statistically, LocalMap ranging errors are indistinguishable from zero in 153 of the 284 ranges considered, and 233 are within 20 cm of zero at the 5% significance level.

Figure 12 plots average LocalMap target vehicle ground speed estimation errors over the 11 parallel head-on encounters. Like the perpendicular intersection encounters, ground speed estimates in the head-on encounters remain accurate at all ranges of the maneuver after initial target acquisition. As in the previous experiment, this accuracy is due to the positioning of the ego vehicle's Delphi radars; one faces forward and measures the speeds of approaching vehicles directly through range rate information in Doppler shifts. Of greater interest is the fact that these accurate speed estimates persist in the closest ranges of the maneuver, as the target vehicle passes to the left of the ego vehicle. At this point in the maneuver the target vehicle is not visible by the forward-facing radar, and its shape as observed by the laser rangefinders varies greatly due to a rapidly changing viewpoint. The LocalMap's point cloud representation suffers no losses from the changing viewpoint, however, and maintains accurate estimates throughout the maneuver. Statistically, LocalMap ground speed estimation errors are indistinguishable from zero at the 5% significance level in 105 of 284 ranges considered in this experiment and within 5 cm / sec. of zero in 246 of the ranges considered.

Figure 13 plots average LocalMap target vehicle relative heading estimation errors over the 11 parallel head-on encounters. Like the ground speed estimation errors, the relative heading estimation errors remain small throughout the entire maneuver. At the 5% significance level, these heading errors are indistinguishable from zero in 93 of 284 ranges considered and within 2° of zero in 266 ranges considered. Like the ground speed errors, the heading errors remain accurate even in the closest ranges of the maneuver, when the target vehicle's shape as observed by the laser rangefinder changes most rapidly. The LocalMap's point cloud representation and parameterized rigid body transform resolve these rapid shape changes correctly and without estimation artifacts present in feature extraction approaches.

C. Experiment 3: Multiple Obstacle Tracking In Heavy Traffic

The third experiment evaluates the LocalMap's consistency in tracking multiple obstacles in a densely-populated, highly dynamic environment. In this experiment, variants of the LocalMap algorithm with different numbers of particles are run on the same segment of logged data, a 19 minute excerpt of a DARPA Urban Challenge qualifying round at George Air Force Base in Victorville, California. The data contains Skynet's sensor measurements of two concentric loops of heavy traffic traveling in opposite directions, recorded as Skynet merged into and out of this traffic multiple times. Figure 14 shows the experimental setup.

The Urban Challenge data excerpt is presented to variants of the LocalMap tracking algorithm run with 1, 2, 5, 10, 20, and 50 particles. No truth data is available for the qualifying round, although the merging scenario features repeated instances of the perpendicular intersection encounters and parallel head-on encounters studied in Sections III-A and III-B. Two new factors are present in this experiment: obstacle occlusion,

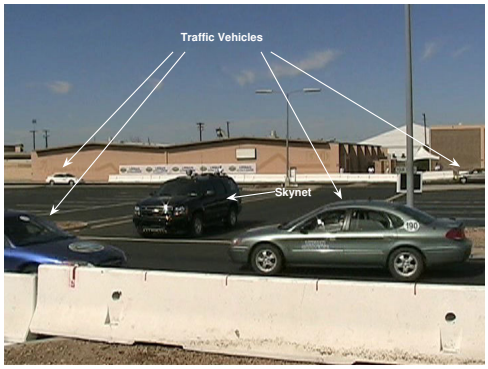

Fig. 14. In the third experiment, variants of the LocalMap are run on excerpts of data from a DARPA Urban Challenge qualifying round. In this data, Skynet autonomously completes multiple merges into and out of moving traffic across a lane of oncoming vehicles.

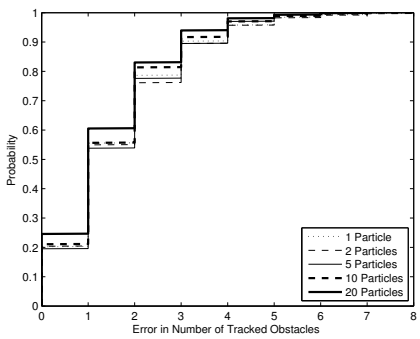

Fig. 15. Sample cumulative distribution function of errors in the number of obstacles tracked in a heavy traffic scenario in variants of the LocalMap run with 1, 2, 5, 10, and 20 particles. Errors are calculated against a LocalMap variant run with 50 particles. The LocalMap algorithm's convergence to a common number of tracked obstacles as the number of particles increases shows stability in selection of data assignments.

where one or more obstacles are temporarily blocked from Skynet's view, and large numbers of moving obstacles. This particular data excerpt features large numbers of occlusions, occurring primarily when a vehicle in the inner traffic loop passes alongside a vehicle in the outer traffic loop. In this environment, the LocalMap tracks an average of 32 potentially moving obstacles at each instant in time.

The close interaction between moving traffic vehicles adds data assignment complexity not present in the first two experiments. Though no truth data is available, the effects of the added complexity are reflected in the number of obstacles tracked by the LocalMap. Incorrect data assignments, even those resolved quickly in resampling, can temporarily result in a particle with too many or too few obstacles. Accurate Monte Carlo sampling of the data assignments, on the other hand, should yield convergence to a common maximum *a posteriori* estimate as the number of particles increases. This experiment looks for that convergence across LocalMap variants as a measure of data assignment consistency in a complex dynamic environment.

Figure 15 plots the sample cumulative distribution function of errors in the number of obstacles tracked in variants of the LocalMap with 1, 2, 5, 10, and 20 particles. Errors are calculated against a LocalMap variant run with 50 particles, and are only evaluated for the maximum *a posteriori* (MAP) estimate in each LocalMap variant, i.e. the particle with the largest weight. Figure 15 shows that the sample cumulative distribution functions for LocalMap variants with 10 and 20 particles are strictly greater than those of variants with fewer particles, indicating convergence to a common number of obstacles as the number of particles increases. More importantly, all the cumulative density functions lie within approximately 5% of each other. From this result it is evident that the LocalMap's point cloud representation and parametric filter representation largely make the data assignments 'obvious,' as even a single particle LocalMap is capable of achieving results similar to a 50 particle variant throughout most of the experiment.

Although a single particle performs almost as well as 50 in most of the experiment, there are three isolated instances where variants with low numbers of particles result in significant errors. Two of these instances, occurring at $t \approx 12.235$ min. and 18.65 min. into the data excerpt, correspond to times when Skynet just starts to merge across oncoming traffic. The scenario is difficult from a data assignment point of view: all traffic vehicles in the oncoming lane instantly become occluded by the closest oncoming vehicle. A metal light pole behind Skynet further compounds the problem, as it temporarily passes in view of Skynet's rear radars at the same point in the maneuver. False positive detections created by a radar speed detector mounted on the light pole could potentially create false obstacles that amplify the discrepancy. The third instance, at $t \approx 7.57$ min., corresponds to Skynet making a $90°$ left turning around a tight corner. Here again Skynet's rear radars are likely the source of the error, as a large construction scissors lift temporarily becomes visible during the turn. This lift likely generates multiple Delphi radar tracks, which may be mistakenly assigned to differing numbers of obstacles in LocalMap variants. All three isolated errors track at most eight fewer or eight more objects than the LocalMap variant with 50 particles. In contrast, Figure 15 shows that most errors are less than four.

IV. Summary Of Performance In The DARPA Urban Challenge

The LocalMap tracking algorithm has also been implemented on Skynet in real-time, where it acts as Skynet's sole obstacle detection and tracking system. The LocalMap is implemented in C++ on a single 2.0 GHz Intel Core 2 Duo machine with 2.0 Gb RAM, running Windows Server 2003. Four particles are used in Skynet's LocalMap to ensure real-time processing. A caching scheme for the measurement function is also implemented to avoid recalculation of $h\left(o_{m,k}, P_k\right)$ when obstacle state and point cloud estimates do not change, and assignment likelihoods are only computed for tracked obstacles in thresholded proximity of a measurement. These minor augmentations help offset the computational expense of numerical differentiation performed in the Sigma Point Transforms utilized in the LocalMap.

Skynet relied on the LocalMap for obstacle detection and tracking during the 2007 DARPA Urban Challenge, a 60 mile

autonomous urban driving competition held in Victorville, California in November, 2007. The Urban Challenge featured simultaneous interaction of 11 full-sized autonomous robots and approximately 50 human-driven sedans in typical urban traffic scenarios such as merging, parking, intersection queuing, and vehicle following. Many of these encounters tested the practical applicability of the LocalMap's point cloud representation and stable measurements. During one qualifying round of the Urban Challenge, for example, Skynet was required to complete multiple merges into and out of moving traffic across a lane of oncoming vehicles. Skynet made two attempts at this qualifying course, completing 5 successful merges into and out of traffic in the first attempt, and 10 in the second attempt, whose data was utilized in Section III-C. In the finals of the Urban Challenge, Skynet was one of only six robots to complete the entire 60 mile course. While competing, Skynet's LocalMap tracked a total of 175252 distinct obstacles. On average, the LocalMap tracked 48.5 obstacles in each particle, with a maximum of 209 obstacles per particle in a single iteration [31].

V. Conclusion

The LocalMap tracking algorithm has been presented as a computationally feasible, real-time solution to the joint estimation problem of data assignment and dynamic obstacle tracking from a potentially moving robotic platform. The algorithm utilizes a Bayesian factorization to separate the joint estimation problem into an independent data assignment problem and a multiple dynamic obstacle tracking problem conditioned on the data assignments. A particle filter is then used to sample the *a posteriori* distribution of data assignments, and compact and efficient parametric filters are used to estimate the *a posteriori* densities of the obstacles conditioned on the sampled data assignments. The LocalMap algorithm achieves a practical computational burden by using expensive Monte Carlo sampling only over the portion of the state space that most needs it, the data assignment histories. The rest of the states, those of the dynamic obstacles, are estimated with banks of efficient closed-form parametric filters.

The LocalMap algorithm achieves real-time rates through a carefully-selected point cloud obstacle representation and stable measurements. These techniques eliminate estimation artifacts and measurement instability common to sensor data preprocessing techniques such as box and corner detectors, improving the accuracy of the LocalMap's parametric tracking filters even under substantial changes in obstacle viewpoint. The result makes data assignments 'obvious,' and real-time performance is achieved through commensurate reductions in particle requirements in the data assignment problem.

The LocalMap algorithm has been implemented on Cornell University's 'Skynet,' an autonomous 2007 Chevrolet Tahoe equipped with a position, velocity and attitude estimator, and laser rangefinders, radars, and optical cameras for obstacle tracking. The LocalMap has been validated in three experiments: two experiments in which a single moving target vehicle is tracked under large changes in obstacle viewpoint, and one experiment in which multiple moving target vehicles are tracked in heavy traffic. In these experiments, the LocalMap is shown capable of both determining the number of obstacles and accurately tracking their positions and velocities. Statistics are also presented for Skynet's performance in the DARPA Urban Challenge, where the LocalMap algorithm was implemented in real-time to serve as Skynet's obstacle detection and tracking system. In the DARPA Urban Challenge, the LocalMap ran for 6 hours, allowing Skynet to travel autonomously for 60 miles in moving traffic.

Acknowledgment

The authors would like to thank the members of the Cornell DARPA Urban Challenge Team for providing sensors, sensor interfaces, and other testing support. The authors would also like to thank Peter Moran and Aaron Nathan for assistance in collecting data. The authors also thank Peter Moran for his photography work.

This work is supported by the DARPA Urban Challenge program (contract no. HR0011-06-C-0147), with Dr. Norman Whitaker as Program Manager.

References

[1] Y. Bar-Shalom and T. Fortmann, *Tracking and Data Association*. Orlando: Academic Press, Inc., 1988.

[2] D. Streller, K. Fürstenberg, and K. Dietmayer, "Vehicle and object models for robust tracking in traffic scenes using laser range images," in *Proceedings of the IEEE 5th International Conference on Intelligent Transportation Systems*, September 2002, pp. 118–123.

[3] S. Thrun, "Robotic mapping: A survey," School of Computer Science, Carnegie Mellon University, Pittsburgh, Tech. Rep. CMU-CS-02-111, February 2002.

[4] C. Wang, "Simultaneous localization, mapping and moving object tracking," PhD Thesis, Robotics Institute, Carnegie Mellon University, April 2004, CMU-RI-TR-04-23.

[5] S. Thrun, Y. Liu, D. Koller, A. Ng, Z. Ghahramani, and H. Durrant-Whyte, "Simultaneous localization and mapping with sparse extended information filters," *The International Journal of Robotics Research*, vol. 23, no. 7-8, pp. 693–716, 2004.

[6] T. Kirubarajan and Y. Bar-Shalom, "Probabilistic data association techniques for target tracking in clutter," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 536–557, March 2004.

[7] Y. Bar-Shalom, T. Kirubarajan, and X. Lin, "Probabilistic data association techniques for target tracking with applications to sonar, radar, and eo sensors," *IEEE A&E Systems Magazine*, vol. 20, no. 8, pp. 37–56, August 2005.

[8] A. Doucet, V. Ba-Ngu, C. Andrieu, and M. Davy, "Particle filtering for multi-target tracking and sensor management," in *Proceedings of the Fifth International Conference on Information Fusion*, vol. 1, 2002, pp. 474–481.

[9] S. Särkkä, A. Vehtari, and J. Lampinen, "Rao-blackwellized monte carlo data association for multiple target tracking," in *Proceedings of the Seventh International Conference on Information Fusion*, P. Svensson and J. Schubert, Eds., vol. 1, Mountain View, CA, 2004, pp. 583–590.

[10] Y. Cheng and T. Singh, "Efficient particle filtering for road-constrained target tracking," in *Proceedings of the Eighth International Conference on Information Fusion*, vol. 1, July 2005, pp. 161–168.

[11] R. Schulz and K. Fürstenberg, "Laserscanner for multiple applications in passenger cars and trucks," in *Advanced Microsystems for Automotive Applications 2006*, 2006, pp. 129–138.

[12] K. Fürstenberg and K. Dietmayer, "Object tracking and classification for multiple active safety and comfort applications using a multilayer laserscanner," in *Proceedings of the 2004 IEEE Intelligent Vehicles Symposium*, June 2004, pp. 802–807.

[13] K. Fürstenberg, K. Dietmayer, and S. Eisenlauer, "Multilayer laserscanner for robust object tracking and classification in urban traffic scenes," in *Proceedings of the Ninth World Congress on Intelligent Transportation Systems*, Chicago, Illinois, 2002, pp. 7–16.

[14] K. Fürstenberg, K. Dietmayer, and V. Willhoeft, "Pedestrian recognition in urban traffic using a vehicle based multilayer laserscanner," in *Proceedings of the 2002 IEEE Intelligent Vehicles Symposium*, vol. 1, June 2002, pp. 31–35.

[15] D. Streller and K. Dietmayer, "Object tracking and classification using a multiple hypothesis approach," in *2004 Intelligent vehicles Symposium*, June 2004, pp. 808–812.
[16] F. Fayad and V. Cherfaoui, "Tracking objects using a laser scanner in driving situation based on modeling target shape," in *Proceedings of the 2007 IEEE Intelligent Vehicles Symposium*, June 2007, pp. 44–49.
[17] R. MacLachlan and C. Mertz, "Tracking of moving objects from a moving vehicle using a scanning laser rangefinder," in *Proceedings of the IEEE Intelligent Transportation Systems Conference 2006*, September 2006, pp. 301–306.
[18] A. Mendes, L. Conde Bento, and U. Nunes, "Multi-target detection and tracking with a laserscanner," in *Proceedings of the 2004 IEEE Intelligent Vehicles Symposium*, June 2004, pp. 796–801.
[19] M. Maehlisch, R. Schweiger, W. Ritter, and K. Dietmayer, "Multisensor vehicle tracking with the probability hypothesis density filter," in *Proceedings of the 9th International Conference on Information Fusion*, Florence, Italy, 2006, pp. 1–8.
[20] M. Maehlisch, T. Kauderer, W. Ritter, and K. Dietmayer, "Feature-level video and multibeam lidar sensor fusion for full-speed acc state estimation," in *Proceedings of the 4th International Workshop on Intelligent Transportation*, 2007.
[21] M. Maehlisch, W. Ritter, and K. Dietmayer, "De-cluttering with integrated probabilistic data association for multisensor multitarget acc vehicle tracking," in *Proceedings of the 2007 IEEE Intelligent Vehicles Symposium*, Istanbul, Turkey, 2007, pp. 178–183.
[22] S. Wender and K. Dietmayer, "3d vehicle detection using a laser scanner and a video camera," in *Proceedings of the 6th European Congress on ITS*, June 2007.
[23] I. Miller and M. Campbell, "Rao-blackwellized particle filtering for mapping dynamic environments," in *Proceedings of the 2007 International Conference on Robotics and Automation*, April 2007, pp. 3862–3869.
[24] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A factored solution to the simultaneous localization and mapping problem," in *Proceedings of the AAAI National Conference on Artificial Intelligence*. Edmonton, Canada: AAAI, 2002.
[25] Y. Bar-Shalom, X. Rong Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software*. New York: John Wiley & Sons, Inc., 2001.
[26] S. Julier and J. Uhlmann, "A new extension of the Kalman filter to nonlinear systems," in *Proceedings of the SPIE Volume 3068: Signal Processing, Sensor Fusion, and Target Recognition VI*, 1997, pp. 182–193.
[27] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd ed. Upper Saddle River: Pearson Education, Inc., 2003.
[28] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, February 2002.
[29] A. Doucet, N. de Freitas, K. Murphy, and S. Russell, "Rao-blackwellised particle filtering for dynamic bayesian networks," in *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, Stanford, 2000, pp. 176–183.
[30] G. Grisetti, C. Stachniss, and W. Burgard, "Improving grid-based slam with Rao-Blackwellized particle filters by adaptive proposals and selective resampling," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2005, pp. 2432–2437.
[31] I. Miller, M. Campbell, D. Huttenlocher, A. Nathan, F.-R. Kline, P. Moran, N. Zych, B. Schimpf, S. Lupashin, E. Garcia, M. Catlin, J. Kurdziel, and H. Fujishima, "Team cornell's skynet: Robust perception and planning in an urban environment," *Journal of Field Robotics*, vol. 25, no. 8, pp. 493–527, August 2008.
[32] I. Miller, B. Schimpf, J. Leyssens, and M. Campbell, "Tightly-coupled gps / ins system design for autonomous urban navigation," in *Proceedings of the 2008 IEEE / ION Position, Localization, and Navigation Symposium*, May 2008.

PLACE PHOTO HERE

Isaac Miller Biography text here.

PLACE PHOTO HERE

Mark Campbell Biography text here.

PLACE PHOTO HERE

Dan Huttenlocher Biography text here.