

SLAM With Dynamic Targets via Single-Cluster PHD Filtering

Chee Sing Lee, Daniel E. Clark, and Joaquim Salvi

Abstract—This paper presents the first algorithm for simultaneous localization and mapping (SLAM) that can estimate the locations of both dynamic and static features in addition to the vehicle trajectory. We model the feature-based SLAM problem as a single-cluster process, where the vehicle motion defines the parent, and the map features define the daughter. Based on this assumption, we obtain tractable formulae that define a Bayesian filter recursion. The novelty in this filter is that it provides a robust multi-object likelihood which is easily understood in the context of our starting assumptions. We present a particle/Gaussian mixture implementation of the filter, taking into consideration the challenges that SLAM presents over target tracking with stationary sensors, such as changing fields of view and a mixture of static and dynamic map features. Monte Carlo simulation results are given which demonstrate the filter's effectiveness with high measurement clutter and non-linear vehicle motion.

Index Terms—Multi-object filtering, SLAM.

I. INTRODUCTION

THE foundations of Simultaneous Localization and Mapping (SLAM) research are commonly attributed to Smith, Self, and Cheeseman [1], who introduced the notion of considering the vehicle and map states as stochastic quantities and applying the Kalman filter or Extended Kalman Filter to produce Bayesian state estimates. In the following decades, other paradigms have arisen for solving the SLAM problem such as those based on optimization [2], [3], or appearance matching [4], [5], however SLAM through Bayesian filtering remains an active line of research [6]–[8].

Until recently little attention has been paid to what we consider to be an important discrepancy in the conceptualization of feature-based SLAM: despite the fact that the map is composed of multiple features, the entire SLAM state has traditionally been treated as a single multivariate probability distribution,

Manuscript received September 09, 2012; revised December 09, 2012; accepted February 19, 2013. Date of publication March 06, 2013; date of current version May 13, 2013. This work was supported by an EPSRC grant EP/J012432/1, EU grant FP7-ICT-2011-7 project PANDORA Ref 288273, Spanish Ministry of Science and Innovation project RAIMON ref. CTM2011-29691-C02-02) and the Catalan Government (FI and BE-DGR grants). The work of C. S. Lee was supported by a Ph.D. FI Scholarship of the Catalan Government. The work of D. E. Clark was supported by an RAEng/EPSCRC Fellowship. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Ba-Ngu Vo.

C. S. Lee and J. Salvi are with the Computer Vision and Robotics Group, University of Girona, Girona 17071, Spain (e-mail: cslee@eia.udg.edu; qsalvi@eia.udg.edu).

D. E. Clark is with the Joint Research Institute in Signal and Image Processing, Heriot-Watt University, Edinburgh EH14 4AS, U.K. (e-mail: d.e.clark@hw.ac.uk).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/JSTSP.2013.2251606

to be estimated as a single object. With such a formulation, measurements are not generated by the entire state vector, but rather a subsection of it. This has prompted the development of additional mechanisms to handle data association such as Joint Compatibility Branch and Bound [9] or 1-Point RANSAC [10]. Furthermore, to introduce new features to the map, the state vector is simply augmented, meaning the dimensionality of the state space constantly fluctuates. The possibility of spurious measurements must be handled through heuristics to decide whether to augment the state vector.

In [11], Mahler proposed Bayesian methods for multi-object estimation. The single random vector of concatenated objects is replaced with a set of individual object vectors, whose cardinality is an uncertain quantity. Working from this premise, the resulting multi-object Bayes filter is one which has the ability to inherently associate measurements with the most likely features, and to separate true measurements from clutter. Tractable first-moment approximations for the multi-object Bayes filter exist in the forms of the PHD filter [12], CPHD filter [13], and CBMeMBer filter [14], which differ based on the assumptions made on prior before the application of Bayes' rule. They have been realized through SMC [15], [16] and Gaussian Mixture [17], [18] implementations, which have been proven to be convergent [19]–[22]. Multi-object Bayes filtering has seen applications in a variety of domains such as sonar [23], millimeter-wave imaging [24], infrared imaging [25], audio signal processing [26], distributed sensing [27], automotive applications [28], [29], and microscopy [30], [31].

To the authors' knowledge, the work of Mullane *et al.* [32], [33] is the first treatment of feature-based SLAM in the context of multi-object filtering. In their work, they propose a Rao-Blackwellized particle filter similar to the FastSLAM family of algorithms [34], where the per-particle Kalman filter banks are replaced with Gaussian Mixture PHD filters. This work has been extended to multi-vehicle SLAM applications [35].

The SLAM algorithm presented in this paper is based on the single-cluster PHD filter. The single-cluster PHD filter was motivated by work on multi-group multi-target tracking [36] and extended object tracking [37]. More general models for processes with interactions extend these models [38], though in this paper we concentrate on the single-cluster PHD filter for modeling the SLAM process. This process, formed as a multi-object process conditioned on a single parent process, has been applied to applications in group tracking [37], sensor registration [39], and calibration of multi-target tracking algorithms [40]. SLAM is another problem which may be modeled as the hierarchical interaction between a parent and daughter process and therefore this filter is applicable. In addition, we demonstrate how this framework may be used to estimate a map which contains

a mixture of stationary and moving features. Algorithms which integrate moving object tracking with SLAM have been proposed in the past, notably by Wang, Thorpe, and Thrun [41]; and by Wang, Sun, and Chiou [42]. Our approach differs in that the dynamic objects contribute to the estimates of the static objects and vehicle pose whereas the previously mentioned works maintain separate posteriors. The following section describes the theoretical formulation of the proposed filter. Section III discusses the filter's implementation and Section IV its validation through simulation. Section V contains a discussion of the results and conclusions.

II. PROBLEM FORMULATION

In this work, we consider a mobile sensor, or vehicle, whose state evolves according to a particular motion model such as constant velocity or coordinated turn. From these motion models, a probability density $\pi(\mathbf{X}_{k|k-1}|\mathbf{X}_{k-1}, \mathbf{u}_k)$, may be derived which describes the probability of transitioning from a previous state \mathbf{X}_{k-1} to a current state $\mathbf{X}_{k|k-1}$, where \mathbf{u}_k is the control input for the current timestep, which may or may not be present depending on the choice of motion model.

Furthermore, we assume that the environment is composed of a set of map features \mathbf{M} . The map is not necessarily completely static, so likewise its state evolves according to $\pi(\mathbf{M}_{k|k-1}|\mathbf{M}_{k-1})$. As the sensor explores the environment, it receives measurements \mathbf{Z}_k relating the map features to the vehicle position through a possibly non-linear function $h(\mathbf{m}, \mathbf{X})$. The measurement process is non-ideal, so that in addition to measurements being subject to additive noise, some map features may fail to be sensed (missed detections), and measurements may be received which do not correspond to any map feature (clutter).

We represent the vehicle state as a single random vector in the vehicle state space $\mathcal{X} \subseteq \mathbb{R}^{n_x}$, and each map feature as a single random vector in the feature space $\mathcal{M} \subseteq \mathbb{R}^{n_m}$.

$$\mathbf{X}_k = [x_{k,1}, \dots, x_{k,n_x}] \in \mathcal{X} \quad (1)$$

$$\mathbf{m}_{k,n} = [m_{k,n,1}, \dots, m_{k,n,n_m}] \in \mathcal{M} \quad (2)$$

While it is certain that there is and always will be only one vehicle, no such certainty applies to the number of map features. Hence the map state is what we call a *Random Finite Set* (RFS) on \mathcal{M} :

$$\mathbf{M}_k = \{\mathbf{m}_{k,1}, \dots, \mathbf{m}_{k,n_k}\} \in \mathcal{F}(\mathcal{M}) \quad (3)$$

where $\mathcal{F}(\mathcal{M})$ denotes the set of all finite subsets of \mathcal{M} . The sensor measurements for each time step are modeled as a union of two independent RFS's on the measurement space $\mathcal{Z} \subseteq \mathbb{R}^{n_z}$: one representing measurements generated by map features, and the other representing the clutter measurements:

$$\mathbf{Z}_k = Z(\mathbf{X}_k, \mathbf{M}_k) \cup C \in \mathcal{F}(\mathcal{Z}) \quad (4)$$

The goal in SLAM is to estimate the joint posterior distribution of the vehicle and map states given a history of control inputs and measurements: $p_k(\mathbf{X}_k) = p_k(\mathbf{X}_k, \mathbf{M}_k|\mathbf{u}_{1:k}, \mathbf{Z}_{1:k})$.

This is accomplished through the prediction and update recursion that is ubiquitous throughout many Bayesian estimation applications, including SLAM. The predicted joint state is given by the Chapman-Kolmogorov equation:

$$p_{k|k-1}(\mathbb{X}_{k|k-1}|\mathbf{u}_{1:k}, \mathbf{Z}_{1:k-1}) = \int \pi(\mathbb{X}_{k|k-1}|\mathbb{X}_{k-1}, \mathbf{u}_k) d\mathbb{X}_{k-1} \quad (5)$$

The updated joint state is the result of applying Bayes' rule:

$$p_k(\mathbb{X}_k|\mathbf{Z}_{1:k}) = \frac{g_k(\mathbf{Z}_k|\mathbb{X}_{k|k-1})p_{k|k-1}(\mathbb{X}_{k|k-1}|\mathbf{u}_{1:k}, \mathbf{Z}_{1:k-1})}{\int g_k(\mathbf{Z}_k|\mathbb{X}_{k|k-1})p_{k|k-1}(\mathbb{X}_{k|k-1}|\mathbf{u}_{1:k}, \mathbf{Z}_{1:k-1}) d\mathbb{X}_{k|k-1}} \quad (6)$$

For all but the most trivial of cases, the above is computationally impractical. Because the joint state is defined over the space $\mathcal{X} \times \mathcal{F}(\mathcal{M})$, the integrals become set integrals [12], which are taken over all sets of all possible cardinalities. Even specifying and storing a probability distribution over such a space becomes a daunting task. These difficulties will be overcome in the following manner:

- 1) The SLAM scenario will be modeled as a cluster process, allowing us to separate the problem into estimation of a relatively simple parent process for the vehicle, and a conditional daughter process for the map.
- 2) Instead of propagating $p_k(\mathbb{X}_k)$, we propagate its first moment $D_k(\mathbb{X}_k)$. The first moment takes values over the single-object space $\mathcal{X} \times \mathcal{M}$ and is significantly easier to handle than a full multi-object probability distribution. This first moment is known variously as the Probability Hypothesis Density (PHD) or intensity of the multi-object distribution.

A. Single-Cluster PHD Filter

This section presents the formulation of the SLAM problem in the context of single-cluster Poisson processes. Point processes [43], or stochastic population processes, are used to describe the location and evolution multiple individuals in some state space, where not only is their distribution over the state space a probabilistic quantity, but the number of targets (cardinality) is as well. Cluster processes, illustrated in Fig. 1, describe a hierarchical arrangement of point processes where the realization of a *daughter* process is conditioned on the realization of some *parent* process [44]. For modeling SLAM, we consider a special subset of cluster processes where the cardinality of the parent process is exactly equal to one. Such processes are termed *single cluster processes*. The map is represented by a daughter point process over the feature space whose realization is conditioned on the realization of the vehicle point process, which consists of a single individual in the vehicle state space.

In this work, both the map and false positive measurements are considered to be realizations of Poisson point processes. This assumption allows us to apply the PHD filter for Poisson single-cluster processes which was derived in [45], and which will be summarized in the following paragraphs.

Under the assumption that the introduction of newly-observed map features is statistically independent of the dynamics

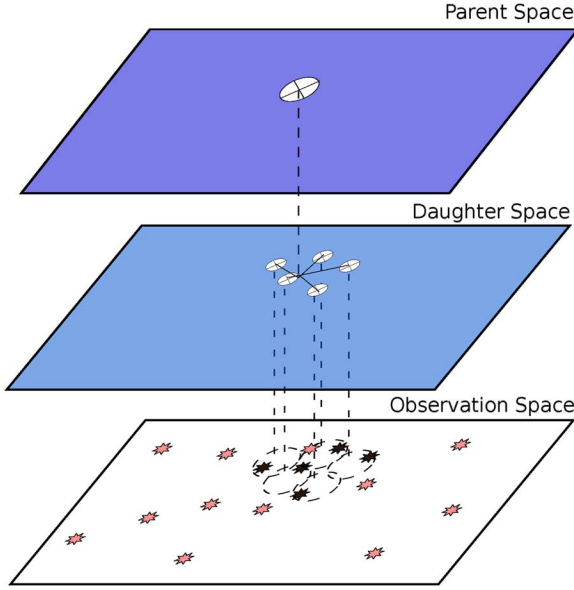


Fig. 1. Visualization of a cluster process. The daughter process is conditioned on realizations of a separate parent process. Observations are collected only on the daughter process, and are mixed in with false alarms.

of existing map features, the predicted joint intensity is given by:

$$D_{k|k-1}(\mathbf{X}, \mathbf{M}) = \int s_{k-1}(\mathbf{X}') \pi_{k|k-1}(\mathbf{X}|\mathbf{X}') \tilde{D}_{k|k-1}(\mathbf{M}|\mathbf{X}') d\mathbf{X}' \quad (7)$$

where $\tilde{D}_{k|k-1}(\mathbf{M}|\mathbf{X})$ is the predicted PHD of the map:

$$\tilde{D}_{k|k-1}(\mathbf{M}|\mathbf{X}) = \gamma_{k|k-1}(\mathbf{M}|\mathbf{X}) + \underbrace{\int \tilde{D}_{k-1}(\mathbf{M}'|\mathbf{X}') p_S(\mathbf{M}'|\mathbf{X}') \pi_{k|k-1}(\mathbf{M}|\mathbf{M}'; \mathbf{X}') d\mathbf{M}'}_{\tilde{D}_{k|k-1}(\mathbf{M}|\mathbf{X})_p} \quad (8)$$

with the following definitions:

$s_{k-1}(\mathbf{X}')$	prior PHD of the vehicle state from time $k-1$
$\pi_{k k-1}(\mathbf{X} \mathbf{X}')$	Markov transition density of the vehicle
$\gamma_{k k-1}(\cdot \mathbf{X}')$	PHD for newly appearing map features at time k , conditioned on vehicle state \mathbf{X}'
$\tilde{D}_{k-1}(\mathbf{M}' \mathbf{X}')$	prior PHD of the map state from time $k-1$, conditioned on vehicle state \mathbf{X}'
$p_S(\cdot \mathbf{X}')$	feature survival probability, conditioned on vehicle state \mathbf{X}'
$\pi_{k k-1}(\cdot \cdot; \mathbf{X}')$	Markov transition density for feature, conditioned on vehicle state \mathbf{X}'
$\tilde{D}_{k k-1}(\mathbf{M} \mathbf{X})_p$	portion of the predict PHD corresponding to persistent map features

The measurement update for the joint PHD is:

$$D_k(\mathbf{X}, \mathbf{M}) = \frac{s_{k|k-1}(\mathbf{X}) L_{\mathbf{Z}_k}(\mathbf{X})}{\int s_{k|k-1}(\mathbf{X}) L_{\mathbf{Z}_k}(\mathbf{X}) d\mathbf{X}} \tilde{D}_k(\mathbf{M}|\mathbf{X}) \quad (9)$$

where $\tilde{D}_k(\mathbf{M}|\mathbf{X})$ is the updated PHD of the map:

$$\begin{aligned} \tilde{D}_k(\mathbf{M}|\mathbf{X}) &= \tilde{D}_{k|k-1}(\mathbf{M}|\mathbf{X}) \\ &\times \left[(1-p_D(\mathbf{M}|\mathbf{X})) + \sum_{z \in \mathbf{Z}_k} \frac{g(z|\mathbf{M}, \mathbf{X}) p_D(\mathbf{M}|\mathbf{X})}{\eta_z(\mathbf{M}|\mathbf{X})} \right] \end{aligned} \quad (10)$$

$$\begin{aligned} \eta_z(\mathbf{X}) &= \kappa_k(z) + \int \tilde{D}_{k|k-1}(\mathbf{M}|\mathbf{X}) p_D(\mathbf{M}|\mathbf{X}) g(z|\mathbf{M}, \mathbf{X}) d\mathbf{M} \end{aligned} \quad (11)$$

with

$s_{k k-1}$	predicted PHD of the vehicle state
$\tilde{D}_{k k-1}(\mathbf{M} \mathbf{X})$	predicted PHD of the map state, conditioned on vehicle state \mathbf{X}
$p_D(\mathbf{M} \mathbf{X})$	detection probability of a feature state \mathbf{M} given a vehicle state \mathbf{X}
$g(z \mathbf{M}, \mathbf{X})$	single-object observation likelihood for a measurement z given a feature state \mathbf{M} and vehicle state \mathbf{X}
$\kappa_k(z)$	PHD of the measurement clutter process evaluated at z
$L_{\mathbf{Z}_k}(\mathbf{X})$	the multi-object observation likelihood of the measurement set \mathbf{Z}_k given a vehicle state \mathbf{X} it is defined by:

$$L_{\mathbf{Z}_k}(\mathbf{X}) = \exp \left\{ - \int p_D(\mathbf{M}|\mathbf{X}) \tilde{D}_{k|k-1}(\mathbf{M}|\mathbf{X}) d\mathbf{M} \right\} \times \prod_{z \in \mathbf{Z}_k} \eta_z(\mathbf{X}) \quad (12)$$

B. Field of View

As the vehicle travels, the observable area of the feature space will change according to the position of the sensor. Let this area be $FOV(\mathbf{X}) \subseteq \mathcal{M}$. We consider map features within $FOV(\mathbf{X})$ to have a probability of detection equal to some nominal value p_D , and all other features to have a zero probability of detection.

$$p_D(\mathbf{M}|\mathbf{X}) = \begin{cases} p_D & \text{if } \mathbf{M} \in FOV(\mathbf{X}) \\ 0 & \text{otherwise} \end{cases} \quad (13)$$

The PHD update in (10) may be thought of as the sum of a non-detection term and a detection term. When $p_D(\mathbf{M}|\mathbf{X}) = 0$ is substituted into (10), the equation reduces to only the non-detection term:

$$\tilde{D}_k(\mathbf{M}|\mathbf{X}) = \tilde{D}_{k|k-1}(\mathbf{M}|\mathbf{X})$$

This means that for the portion of the map that is outside of the vehicle's field of view, the updated PHD is the same as the predicted PHD. Therefore, the PHD update only needs to be performed on map features which are currently visible, which introduces an upper bound for the required computational effort in this portion of the algorithm.

Clearly, this is a simplified model; for application in real-world systems, a feature's probability of detection could be affected by its range from the sensor, or by occlusion. The effects of mismatched detection probabilities should not be discounted. If a feature is assumed to be in the FOV when in reality it is not, its p_D will be overestimated. This means the non-detection term in (10) will be under-weighted, and because no measurement was received, the detection term will also have a very low weight. Depending on the state extraction scheme used, this may result in a map feature being incorrectly deleted. If the FOV proves too challenging to be parameterized, then it may be worthwhile to incorporate techniques for online estimation of the detection probability [46].

C. Dynamic Map Features

Traditionally, feature-based SLAM solutions operate under the assumption that the map consists only static features. There may be transient objects moving through the environment which are detected by the vehicle's sensors, but these measurements are excluded through some pre-processing step in order to fit the original assumption. We hypothesize that the SCPHD-SLAM framework is capable of discriminating between static and dynamic features, and of maintaining tracks for dynamic features, even when sensor measurements provide no information on the static or dynamic nature of the features. We can define a state space with both position and velocity components. For example in the Cartesian plane:

$$\mathbf{m} = [x, y, \dot{x}, \dot{y}] \subseteq \mathbb{R}^4$$

One might expect that given this state definition, we could model the feature evolution with some appropriate motion model, e.g. constant velocity or coordinated turn, and estimates for static features would conveniently converge towards points on the hyperplane $(\dot{x}, \dot{y}) = (0, 0)$. Unfortunately, this does not work as well as anticipated. The Markov transition densities formed from these motion models propagate the uncertainty in the velocity estimate as uncertainty in the position estimate, so that even in the absence of process noise, the position uncertainty grows every time step during the prediction step unless the feature's velocity is known deterministically. This has serious implications for features outside of the field of view, whose position uncertainties will continue to grow until reobserved. To address these issues, we instead represent the daughter process as the sum of two Poisson processes, one for static features, and one for dynamic features

$$\tilde{D}(\mathbf{M}|\mathbf{X}) = \tilde{D}_s(\mathbf{M}|\mathbf{X}) + \tilde{D}_d(\mathbf{M}|\mathbf{X}) \quad (14)$$

Because the sum of Poisson processes is still Poisson, the previously described equations are still applicable. We consider the positions for static features to reside in a different space than the positions of the dynamic features, so that \mathbf{M} is now defined over the joint space $\mathcal{M} = \mathcal{M}_s \times \mathcal{M}_d$.

The prediction step will be performed separately for each process, according to (8), using their particular Markov transition density and survival probability.

D. Multi-Object Likelihood

In this section we consider the difference between our solution and the Rao-Blackwellized PHD filter based solution proposed by Mullane, Vo and Adams [32] (RBPHD-SLAM). The most important difference is the derivation of the multi-object likelihood $L_{\mathbf{Z}_k}(\mathbf{X})$, which is referred to as $g(\mathbf{Z}_k|\mathbf{X})$ in their work. The RBPHD-SLAM likelihood is obtained by observing that it is equivalent to the normalization term in the daughter update. The Bayes' update for the conditional map is:

$$p_{k|k}(\mathbf{M}|\mathbf{X}) = \frac{g(\mathbf{Z}_k|\mathbf{M}, \mathbf{X})p_{k|k-1}(\mathbf{M}|\mathbf{X})}{g(\mathbf{Z}_k|\mathbf{X})} \quad (15)$$

This can be re-arranged to become:

$$g(\mathbf{Z}_k|\mathbf{X}) = \frac{g(\mathbf{Z}_k|\mathbf{M}, \mathbf{X})p_{k|k-1}(\mathbf{M}|\mathbf{X})}{p_{k|k}(\mathbf{M}|\mathbf{X})} \quad (16)$$

Because the $g(\mathbf{Z}_k|\mathbf{X})$ has no dependence on the map, any choice for the value of \mathbf{M} will produce the same likelihood. Therefore, it is convenient to assume an empty map ($\tilde{\mathbf{M}} = \emptyset$) or a single-feature map ($\tilde{\mathbf{M}} = \{m\}$) for ease of computation. To obtain a closed-form expression for $g(\mathbf{Z}_k|\mathbf{X})$, the distributions $p_{k|k-1}$ and $p_{k|k}$ are assumed to be Poisson, and their first moments are evaluated at $\tilde{\mathbf{M}}$ rather than the true distribution.

The single-cluster PHD filter was derived with the approach proposed by Mahler [12], where an assumption on the prior is used to derive a closed-form expression from the exact Bayes filter update. Specifically, we assume that the prior is the realization of a hierarchical Poisson process. The approach by Mullane, Vo and Adams required Poisson approximations on both the prior and posterior, and a further approximation on the number of features. Consequently, the single-cluster PHD filter more faithfully represents the true multi-object distribution since it requires fewer approximations in its derivation.

III. IMPLEMENTATION

In implementing the SC-PHD SLAM filter, we couple a particle representation for the vehicle state with a Gaussian mixture representation for the PHD of the map state. To reflect the conditional relationship between the parent and daughter processes, each particle in the vehicle PHD is associated with its own Gaussian Mixture representing the map estimate conditioned on that particle's trajectory.

$$s_{k-1}(\mathbf{X}) = \sum_{i=1}^{N_{k-1}} \zeta_{k-1}^{(i)} \delta(\mathbf{X} - \mathbf{X}_{k-1}^{(i)}) \quad (17)$$

$$D_{k-1}^{(i)}(\mathbf{M}|\mathbf{X}_{k-1}) = \sum_{j=1}^{J_{k-1}^{(i)}} w^{(i,j)} \mathcal{N}(\mathbf{M}; \boldsymbol{\mu}_{k-1}^{(i,j)}, \mathbf{P}_{k-1}^{(i,j)}) \quad (18)$$

Note that because the PHD is not a true probability distribution, the weights $w^{(i,1)}, \dots, w^{(i,j)}$ do not necessarily sum to 1. Rather, the sum of these weights is the estimated number of map features. The particle representation was chosen for its ability to capture non-linearities in vehicle motion, while the

Gaussian mixture representation was chosen for its computational economy and was deemed suitable under the assumption of relatively simple dynamics for map features. Each particle's map is propagated with a GM-PHD filter [17], amended with the measurement-driven birth strategy described in [47]. The GM-PHD filter relies on the further assumptions that the map feature motion model and measurement models are linear Gaussian, or may be reasonably linearized as such, with matrices \mathbf{F} and \mathbf{H} , and that the noise affecting these models is zero-mean Gaussian-distributed, with covariance matrices \mathbf{Q} and \mathbf{R} .

A. Prediction

Substitution of (17) and (18) into the SC-PHD prediction (7), and applying the sampling property of the Dirac delta function yields the following result:

$$D_{k|k-1}(\mathbf{X}, \mathbf{M}) = \sum_{i=1}^{N_{k-1}} \zeta_{k-1}^{(i)} \pi_{k|k-1}(\mathbf{X}|\mathbf{X}_{k-1}^{(i)}) \tilde{D}_{k|k-1}(\mathbf{M}|\mathbf{X}_{k-1}^{(i)}) \quad (19)$$

Next, the Markov transition density of the parent is approximated by drawing M samples from it:

$$\{\mathbf{X}^{(i,1)}, \dots, \mathbf{X}^{(i,M)}\} \sim \pi_{k|k-1}(\mathbf{X}|\mathbf{X}_{k-1}^{(i)}) \quad (20)$$

$$D_{k|k-1}(\mathbf{X}, \mathbf{M}) \approx \sum_{i=1}^{N_{k-1}} \sum_{j=1}^M \frac{\zeta^{(i)}}{M} \delta(\mathbf{X} - \mathbf{X}^{(i,j)}) \tilde{D}_{k|k-1}(\mathbf{M}|\mathbf{X}_{k-1}^{(i)}) \quad (21)$$

With measurement-driven births, the incorporation of new targets into the map PHD is delayed until the update step. Therefore, the predicted map PHD describes only persisting map features propagated forward in time:

$$\tilde{D}_{S,k|k-1}(\mathbf{M}|\mathbf{X}_{k-1}^{(i)}) = p_S \sum_{j=1}^{J_{S,k|k-1}} w_{S,k|k-1}^{(j)} \mathcal{N}(\mathbf{M}; \boldsymbol{\mu}_{S,k|k-1}^{(j)}, \mathbf{P}_{S,k|k-1}^{(j)}|\mathbf{X}) \quad (22)$$

$$w_{S,k|k-1}^{(j)} = w_{k-1}^{(j)} \quad (23)$$

$$\boldsymbol{\mu}_{S,k|k-1}^{(j)} = f(\boldsymbol{\mu}_{k-1}^{(j)}) \quad (24)$$

$$\mathbf{P}_{S,k|k-1}^{(j)} = \mathbf{F}_k \mathbf{P}_{k-1}^{(j)} \mathbf{F}_k^T + \mathbf{W}_k \mathbf{Q}_k \mathbf{W}_k^T \quad (25)$$

where $f(\mu)$ is the (possibly non-linear) motion model describing the motion of map features and \mathbf{F} and \mathbf{W} are the Jacobians of the motion model with respect to the map feature and process noise respectively. It should be noted that for static map features, the motion model is simply the identity, and so the prediction only needs to be performed for dynamic map features. The predicted joint PHD now consists of $N_{k|k-1} = M \times N_{k-1}$ particles corresponding to the vehicle state, and the same number of Gaussian mixtures representing the conditional map PHDs. Note that this means the number of particles grows geometrically with every prediction, but when

the particles are resampled, we only draw a number of samples equal to the number of original particles, so that this growth does not continue unchecked.

B. Update

Each of the conditional map PHDs are updated individually when measurement inputs arrive. As mentioned previously, the incorporation of new targets into the PHD is postponed until the measurement update. Thus, the updated PHD can be considered as the sum of three terms: 1) Targets persisting from the previous time step which have failed to be detected (mis-detection); 2) Targets persisting from the previous time step which have been detected (detection); and 3) Newly-appearing targets for the current timestep (birth). We assume that the static/dynamic nature of a map feature is not captured by the sensor, so for each measurement, we create two birth terms: one in the static feature space and one in the dynamic feature space.

$$\begin{aligned} \tilde{D}_{k|k}(\mathbf{M}|\mathbf{X}) = & (1-p_D) \sum_{j=1}^{J_{k|k-1}} w_{k|k-1}^{(j)} \mathcal{N}(\mathbf{M}; \boldsymbol{\mu}_{k|k-1}^{(j)}, \mathbf{P}_{k|k-1}^{(j)}) \\ & + \sum_{z \in \mathbf{Z}_k} \sum_{j=1}^{J_{k|k-1}} w_{k|k}^{(j)} \mathcal{N}(\mathbf{M}; \boldsymbol{\mu}_{k|k}^{(j)}(z), \mathbf{P}_{k|k}^{(j)}(z)) \\ & + \sum_{z \in \mathbf{Z}_k} \frac{w_0^\gamma}{\eta_z(\mathbf{M}|\mathbf{X})} \\ & \times (\mathcal{N}(\mathbf{M}; z_s^*(\mathbf{X}), \mathbf{R}_s^*) + \mathcal{N}(\mathbf{M}; z_d^*(\mathbf{X}), \mathbf{R}_d^*)) \end{aligned} \quad (26)$$

$$w_{k|k}^{(j)} = w_{k|k-1}^{(j)} \frac{p_D g(z|\mathbf{M}, \mathbf{X}) w_{k|k-1}^{(j)}}{\eta_z(\mathbf{M}|\mathbf{X})} \quad (27)$$

$$\eta_z(\mathbf{M}|\mathbf{X}) = \kappa(z) + p_D \sum_{l=1}^{J_{k|k-1}} g(z|\mathbf{M}, \mathbf{X}) w_{k|k-1}^{(l)} + 2w_0^\gamma \quad (28)$$

$$\boldsymbol{\mu}_{k|k}^{(j)}(z) = \boldsymbol{\mu}_{k|k-1}^{(j)} + \mathbf{K} \left(z - h(\boldsymbol{\mu}_{k|k-1}^{(j)}, \mathbf{X}) \right) \quad (29)$$

$$\mathbf{P}_{k|k}^{(j)}(z) = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1}^{(j)} \quad (30)$$

$$\mathbf{K}_k = \mathbf{P} \mathbf{H}_k^T \mathbf{S}_k^{-1} \quad \mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k \quad (31)$$

$z_s^*(\mathbf{X})$ and $z_d^*(\mathbf{X})$ are the static and dynamic map feature states obtained by inverting the measurement z using vehicle state \mathbf{X} , with corresponding covariances $\mathbf{R}_s^*(\mathbf{X})$ and $\mathbf{R}_d^*(\mathbf{X})$. A reasonable value for R^* is

$$\mathbf{R}_{(\cdot)}^*(\mathbf{X}) = \mathbf{J}^* \mathbf{R}_{(\cdot)} \mathbf{J}^{*T} \quad (32)$$

where \mathbf{J}^* is the Jacobian of the inverse measurement function with respect to the measurement. These feature states will have identical position components, and will differ only in the velocity components. The weights of the parent particles are updated using the multi-object measurement likelihood $L_{\mathbf{Z}_k}$.

$$\zeta_k^{(i)} = \frac{L_{\mathbf{Z}_k}(\mathbf{X}_{k|k-1}^{(i)}) \zeta_{k|k-1}^{(i)}}{\sum_{l=0}^{N_{k|k-1}} L_{\mathbf{Z}_k}(\mathbf{X}_{k|k-1}^{(l)}) \zeta_{k|k-1}^{(l)}} \quad (33)$$

$$L_{\mathbf{Z}_k}(\mathbf{X}_{k|k-1}^{(i)}) = \exp \left\{ \sum_{j=0}^{J_{k|k-1}^{(i)}} w_{k|k-1}^{(j)} \right\} \times \prod_{z \in \mathbf{Z}_k} \eta_z(\mathbf{M}_{k|k-1}^{(i)} | \mathbf{X}_{k|k-1}^{(i)}) \quad (34)$$

Following each map update, the number of terms in the Gaussian mixture is equal to $J_{k|k} = (|\mathbf{Z}_k| + 1)J_{k|k-1} + |\mathbf{Z}_k|$. To limit this exponential growth in mixture terms, a mixture reduction step is performed to eliminate terms with weights that are too low, and to merge similar terms. Specifically, we employ the clustering algorithm described in [48], but an alternative reduction algorithm, such as Gaussian Mixture Reduction by Clustering [49] may be substituted here if higher fidelity is required.

C. State Extraction

To obtain an estimate of the vehicle state, the weighted mean of the parent particles is used. For the map, we select the PHD corresponding to the maximum-weighted particle. An equally-valid approach would be to compute the weighted mean of all the particles' PHDs, but this was not done for computational considerations. Next, the Gaussian mixture weights are summed to give an estimate of the number of targets N_k . The N_k most highly-weighted terms are then taken to be the map estimate.

IV. SIMULATION RESULTS

The SC-PHD SLAM filter was verified via synthetic simulation using the scenario depicted in Fig. 2(a). In this scenario, a vehicle explores a planar two-dimensional environment, and receives measurements generated by point features, several of which are themselves moving through the environment. The vehicle state is represented by a three element vector consisting of absolute displacement in x and y , and orientation θ : $\mathbf{X}_k = [x_k, y_k, \theta_k]^T$. The dynamics of the vehicle adhere to the motion model described in [50], [51], where the next vehicle state depends on control inputs for velocity v_e taken from a wheel encoder, and steering angle α .

$$\begin{aligned} [x_k, y_k, \theta_k]^T &= [x_{k-1}, y_{k-1}, \theta_{k-1}]^T \\ &+ \begin{bmatrix} \dot{x} + \dot{\theta}(a \sin \theta_{k-1} + b \cos \theta_{k-1}) \\ \dot{y} + \dot{\theta}(a \cos \theta_{k-1} - b \sin \theta_{k-1}) \\ \dot{\theta} \end{bmatrix} \Delta T \end{aligned} \quad (35)$$

$$\dot{x} = v_e \cos \theta_{k-1} \quad \dot{y} = v_e \sin \theta_{k-1} \quad \dot{\theta} = \frac{v_e}{L} \tan \alpha \quad (36)$$

where $a = 1.890$ m, $b = 0.250$ m, $H = 0.380$ m, and $L = 1.415$ m are parameters describing vehicle geometry and sensor placement, also

$$v_c = \frac{v_e}{1 - \tan \alpha \frac{H}{L}} \quad (37)$$

transforms wheel encoder velocity onto the vehicle centerline. The odometry inputs are reported with zero-mean additive

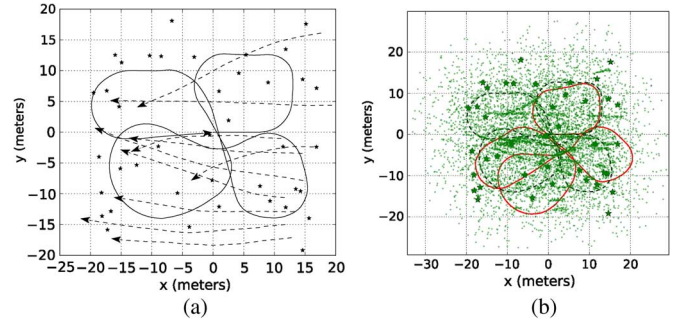


Fig. 2. Simulation scenario. (a) Vehicle trajectory (solid line), static and dynamic map features (stars and dashed lines); (b) cumulative measurements and dead-reckoning trajectory (red, solid).

Gaussian noise with standard deviations $\sigma_v = 1$ m/s and $\sigma_\alpha = 2^\circ$ for wheel velocity and steering angle respectively.

The measurements received by the vehicle from environment features consist of a range r and a relative bearing ϕ .

$$\begin{bmatrix} r_k \\ \phi_k \end{bmatrix} = \begin{bmatrix} \sqrt{\Delta x^2 + \Delta y^2} \\ \arctan(\Delta y, \Delta x) \end{bmatrix} \quad (38)$$

The vehicle's sensor has a 15 m range, and a 360° field of view. Measurements are subject to zero-mean additive Gaussian noise, with standard deviations $\sigma_r = 0.25$ m and $\sigma_\phi = 0.5^\circ$ for range and bearing respectively. Moreover, false alarm measurements are received at a mean rate of $\lambda = 20$ per scan. Odometry inputs arrive at a rate of 20 Hz, while measurements arrive at a rate of 10 Hz. The probability of detection was $p_D = 0.95$. Fig. 2(b) gives an example of the dead-reckoning trajectory from the noisy odometry inputs, and cumulative sensor measurements which may be generated from these simulation parameters.

Monte Carlo (MC) simulations were performed with 25 different sets of odometry and measurement inputs. To account for variability in sampling the Markov transition during the prediction, 4 runs were performed for each set of inputs, for a total of 100 MC runs. The filter was run with a nominal particle count of $N_0 = 32$, and $M = 2$ samples were drawn from the transition density every prediction step. The threshold for particle resampling was set to $N_{eff} = 0.05$. Fig. 3(a) shows an example of the resulting map and trajectory estimates. Dynamic feature tracks are shown in Fig. 3(b). It can be seen that the majority of false tracks are located where there are closely-spaced static map features, and where successive measurements originating from these features could be construed as originating from a single dynamic feature.

Figs. 3(c) and 3(d) show the average error in vehicle pose and map. Mapping performance was quantified using the OSPA distance [52], a metric which takes into account differences in both localization and cardinality. For computing the map error, the ground truth was constructed in the following manner: 1) Static features are included in the set of true features from the time step when they are first observed until the end of the simulation; and 2) Dynamic features are included in the set of true features only while they are within the field of view; this results

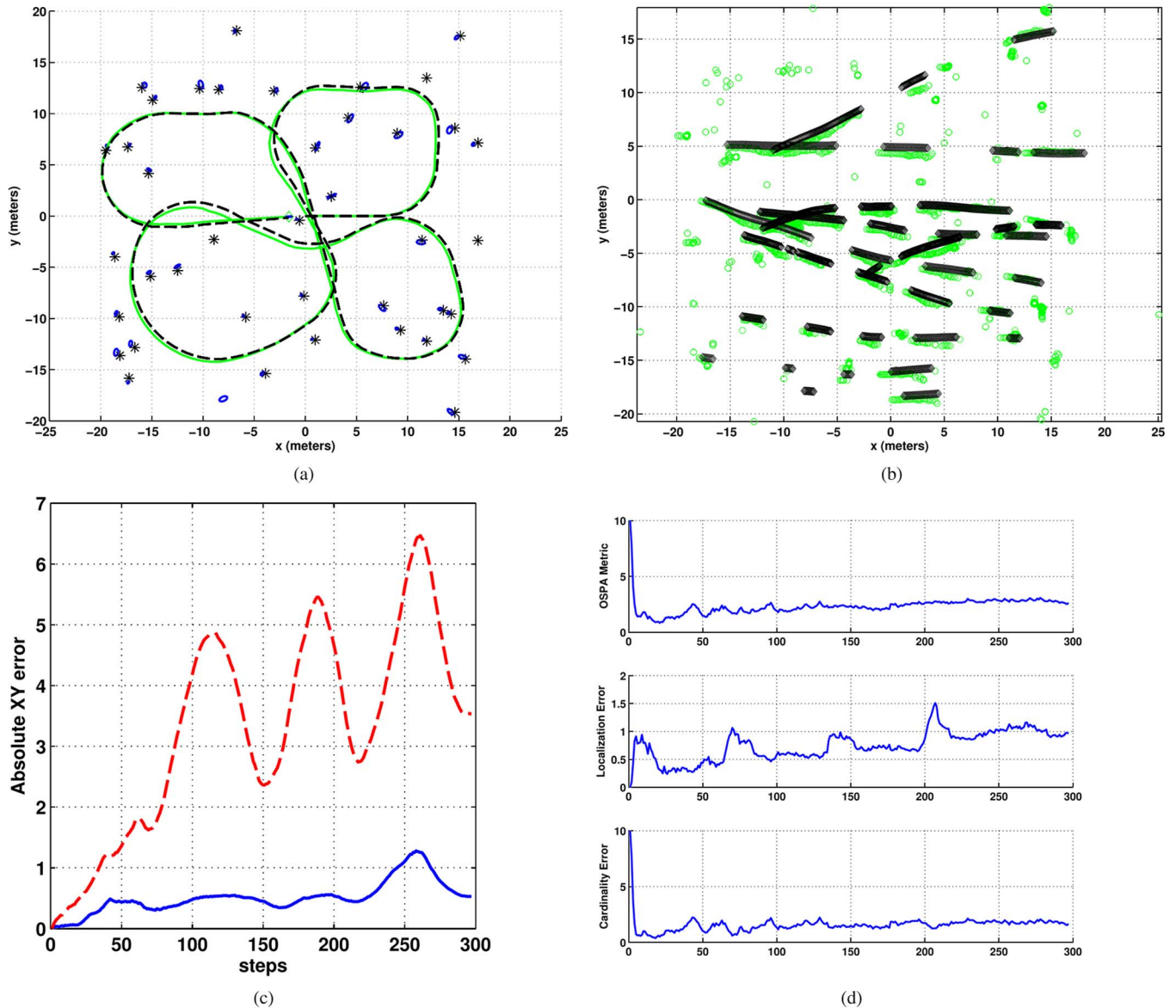


Fig. 3. Simulation results. (a) Example map and trajectory estimate. Black stars and dashed line show true feature locations and vehicle trajectory; (b) tracks of estimated dynamic targets (green), overlaid on true feature trajectories (black); (c) mean vehicle error over 100 MC runs (blue, solid) and dead-reckoning error (red, dashed); (d) mean map error over 100 MC runs. A cutoff value of $c = 10$ was used in computing the OSPA metric.

in the segmented ground truth tracks in Fig. 3(b). The results indicate that the SCPHD SLAM algorithm produces a definitively superior trajectory estimate compared to odometry alone, and that map estimation error is non-increasing for the duration of the simulation. For the sake of expeditious computation in our MC runs, we used a reduced-size scenario of approximately 300 time steps in length. However, in order to ascertain the algorithm’s viability in more extended operations, we also ran a single simulation on a scenario 3500 time steps in length, with identical parameters. The results are shown in Figs. 4, and show that consistent estimation performance is sustained over longer running times. As each per-particle map is updated independently, the SC-PHD SLAM algorithm lends itself well to parallelization. Our implementation is based on the CUDA 4.0 parallel computing platform, and the simulations presented here were performed on an Nvidia Tesla C2070 GPU.

V. DISCUSSION

The results demonstrate that the single-cluster PHD filter is suitable to Simultaneous Localization and Mapping applications. We believe that its ability to cope with measurement clutter without the need for data association is an attractive property of the SC-PHD filter, and with further development, it could be successfully applied to real-world scenarios. In addition, this framework can correctly discriminate and track dynamic map features mixed in with static features, even when the sensor measurements provide no information about the movement of the feature. Whereas in the past moving objects in the environment have been regarded as a nuisance for SLAM algorithms, we have shown that they can in fact be exploited for localization purposes.

Nevertheless, there is room for improvement. We anticipate that the ambiguity between dynamic map features and

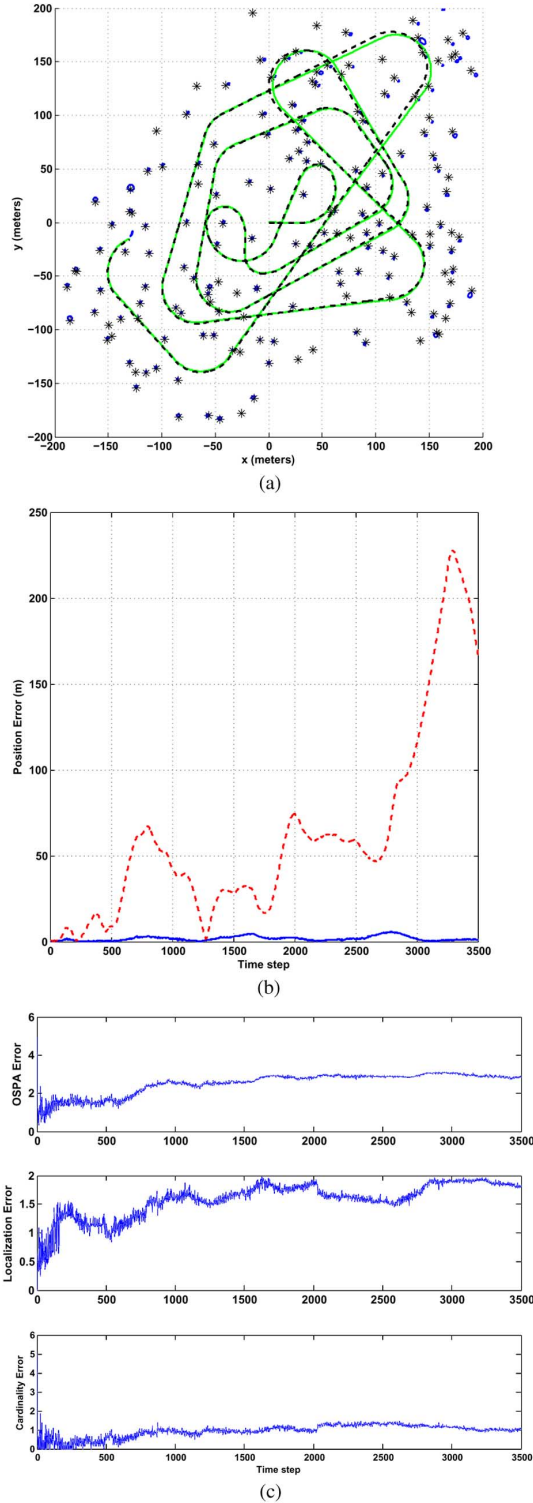


Fig. 4. Results from 3500 timestep simulation. (a) Trajectory and map results. Black stars and dashed line show true feature locations and vehicle trajectory; (b) vehicle pose estimation error (blue, solid), compared to error from odometry alone (red, dashed); (c) map estimation OSPA error, with localization and cardinality components.

closely-spaced static features can be resolved by introducing more complex birth models for the dynamic features. Like other particle filter SLAM algorithms such as FastSLAM and RBPHD-SLAM, the SCPHD-SLAM filter has a tendency towards degeneracy in the vehicle state estimate, but this could be

Input: : Prior state estimate, current control inputs and/or current measurements

Output: : Posterior state estimate, incorporating controls and measurements

procedure SCPHDSLAM($D_{k-1}, \mathbf{u}_k, \mathbf{Z}_k$)

for $i = 1 \dots N_{k-1}$ **do**

$D_{k|k-1}^{(i)} = \text{PREDICT}(D_{k-1}^{(i)}, \mathbf{u}_k)$

$[\mu_+^{(i)}, \mathbf{P}_+^{(i)}] = \text{PREUPDATE}(D_{k|k-1}^{(i)}, \mathbf{Z}_k)$

$[\mathbf{M}_k^{(i)}, L\mathbf{Z}_k] = \text{UPDATE}(D_{k|k-1}^{(i)}, \mu_+^{(i)}, \mathbf{P}_+^{(i)}, \mathbf{Z}_k)$

$\zeta_k^{(i)} = \zeta_{k|k-1}^{(i)} \times L\mathbf{Z}_k$

$\text{PRUNEANDMERGE}(\mathbf{M}_{s,k}^{(i)})$ // [17, Table II]

$\text{PRUNEANDMERGE}(\mathbf{M}_{d,k}^{(i)})$

end for

// Particle resampling according to [53, Algorithm 2]

$\mathbf{X}_k^{(1\dots N_k)} = \text{RESAMPLE}(\mathbf{X}_{k|k-1}^{(1\dots N_{k|k-1})}, \zeta_k^{(1\dots N_{k|k-1})})$

$\zeta_k^{(1\dots N_k)} = 1/N_k$

return $D_{k|k} = \{\mathbf{X}_k^{(1\dots N_k)}, \mathbf{M}^{(1\dots N_k)}, \zeta_k^{(1\dots N_k)}\}$

end procedure

Fig. 5. SC-PHD SLAM algorithm overview.

Input: Prior vehicle particles and map PHD, control input

Output: Predicted vehicle and map PHD

procedure PREDICT($D_{k-1}(\mathbf{X}^{(i)}, \mathbf{M}^{(i)}), \mathbf{u}_k$)

$\mathbf{X}_{k|k-1}^{(1\dots M)} \sim \pi(\mathbf{X}|\mathbf{u}_k, \mathbf{X}_{k-1}^{(i)})$ // Sample transition density

$\mathbf{M}_{k|k-1}^{(1\dots M)} \leftarrow \mathbf{M}_{k-1}^{(i)}$ // duplicate maps

$\eta_{k|k-1}^{(1\dots M)} \leftarrow \eta_{k-1}^{(i)}/M$

// Predict map features

for all $\{\mu_{k-1}^{(j)}, \mathbf{P}_{k-1}^{(j)}, w_{k-1}^{(j)}\}$ in $\mathbf{M}_{k|k-1}^{(1\dots M)}$ **do**

$w_{k|k-1}^{(j)} = w_{k-1}^{(j)}$

$\mu_{k|k-1}^{(j)} = f(\mu_{k-1}^{(j)}, \mathbf{u}_k)$

$\mathbf{P}_{k|k-1}^{(j)} = F_k^{(j)} \mathbf{P}_{k-1}^{(j)} F_k^{(j)T} + W_k^{(j)} Q_k^{(j)} W_k^{(j)T}$

end for

return $\{\mathbf{X}_{k|k-1}^{(1\dots M)}, \mathbf{M}_{k|k-1}^{(1\dots M)}, \eta_{k|k-1}^{(1\dots M)}\}$

end procedure

Fig. 6. SC-PHD SLAM per-particle prediction.

Input: Predicted particle and map, measurements

Output: Pre-update terms for SC-PHD daughter update

procedure PREUPDATE($D_{k|k-1}(\mathbf{X}^{(i)}, \mathbf{M}^{(i)}), \mathbf{Z}_k$)

for all $\{\mu^{(j)}, \mathbf{P}^{(j)}, w^{(j)}\}$ **do**

$\hat{z} = h^{-1}(\mathbf{X}, \mu^{(j)})$ // Predicted measurement

$\mathbf{S} = \mathbf{H}\mathbf{P}^{(j)}\mathbf{H}^T + \mathbf{R}$ // Innovation covariance

$\mathbf{K} = \mathbf{P}^{(j)}\mathbf{H}^T\mathbf{S}^{-1}$ // Kalman Gain

$\mathbf{P}_+^{(j)} = (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P}$ // Updated covariance

for $m = 1 \dots |\mathbf{Z}_k|$ **do**

// single-object likelihood

$g(z_m|\mu^{(j)}, \mathbf{X}) = \mathcal{N}(z; \hat{z}, \mathbf{S})$

$\mu_+^{(j,m)} = \mu + \mathbf{K}(z - \hat{z})$ // Updated mean

end for

end for

return $\{\mu_+^{(1\dots J)}, \mathbf{P}_+^{(1\dots J)}, g(\cdot|\cdot, \mathbf{X})\}$

end procedure

Fig. 7. SC-PHD SLAM per-particle pre-update.

ameliorated through the use of techniques such as particle filter regularization. Furthermore, if the non-linearity of the vehicle motion is not significant, then Gaussian implementations for the parent filter are also possible. It may be worthwhile to explore alternative modelings of the SLAM process. The Poisson

Input: Predicted PHD and pre-update terms

Output: Updated PHD

```

procedure UPDATE( $D_{k|k-1}(\mathbf{X}^{(i)}, \mathbf{M}^{(i)})$ ,  $\mu_+$ ,  $\mathbf{P}_+$ ,  $g(\cdot|\cdot, \mathbf{X}^{(i)})$ )
  for all  $\mu_{k|k-1}^{(j)}$ ,  $\mathbf{P}_{k|k-1}^{(j)}$ ,  $w_{k|k-1}^{(j)}$  do
    // Non-detection terms
     $\mu_{nd}^{(j)} = \mu_{k|k-1}^{(j)}$ ;  $\mathbf{P}_{nd}^{(j)} = \mathbf{P}_{k|k-1}^{(j)}$ ;  $w_{nd} = w_{k|k-1}^{(j)}$ 
    // Detection terms
    for  $m = 1 \dots |\mathbf{Z}_k|$  do
       $\mu_d^{(j,m)} = \mu_+^{(j,m)}$ ;  $\mathbf{P}_d^{(j,m)} = \mathbf{P}_+^{(j,m)}$ ;
       $w_d^{(j,m)} = w_{k|k-1} \mathcal{P}Dg(z_m | \mu^{(j)}, \mathbf{X})$ 
    end for
    // Static and dynamic birth terms
    for  $m = 1 \dots |\mathbf{Z}_k|$  do
       $\mu_{b,s}^{(m)} = z_s^*(\mathbf{X})$ ;  $\mathbf{P}_{b,s}^{(m)} = \mathbf{R}_s^*$ ;  $w_{b,s}^{(m)} = w_b$ 
       $\mu_{b,d}^{(m)} = z_d^*(\mathbf{X})$ ;  $\mathbf{P}_{b,d}^{(m)} = \mathbf{R}_d^*$ ;  $w_{b,d}^{(m)} = w_b$ 
    end for
    // Normalize weights,
    // compute multi-object likelihood
     $N_{k|k-1} = \sum_{j=1}^J w_{k|k-1}^{(j)}$ 
     $L_{\mathbf{Z}_k} = \exp(N_{k|k-1})$ 
    for  $m = 1 \dots |\mathbf{Z}_k|$  do
       $\eta_{z_m} = \kappa(z_m) + \sum_{j=1}^J w_d^{(j,m)} + 2w_b$ 
      for all  $w \in \{w_d^{(1\dots J,m)}, w_{b,s}^{(m)}, w_{b,d}^{(m)}\}$  do
         $w = w / \eta_{z_m}$  // Note: in-place modification
      end for
       $L_{\mathbf{Z}_k} = L_{\mathbf{Z}_k} \times \eta_{z_m}$ 
    end for
  end for
  // Concatenate terms
   $\mu_k = [\mu_{nd}^{(1\dots J)}, \mu_d^{(1\dots J, k|k-1, 1\dots |\mathbf{Z}_k|)}, \mu_{b,s}^{(1\dots |\mathbf{Z}_k|)}, \mu_{b,d}^{(1\dots |\mathbf{Z}_k|)}]$ 
   $\mathbf{P}_k = [\mathbf{P}_{nd}^{(1\dots J)}, \mathbf{P}_d^{(1\dots J, k|k-1, 1\dots |\mathbf{Z}_k|)}, \mathbf{P}_{b,s}^{(1\dots |\mathbf{Z}_k|)}, \mathbf{P}_{b,d}^{(1\dots |\mathbf{Z}_k|)}]$ 
   $w_k = [w_{nd}^{(1\dots J)}, w_d^{(1\dots J, k|k-1, 1\dots |\mathbf{Z}_k|)}, w_{b,s}^{(1\dots |\mathbf{Z}_k|)}, w_{b,d}^{(1\dots |\mathbf{Z}_k|)}]$ 
  return  $\{\mu_k^{(1\dots J_k)}, \mathbf{P}_k^{(1\dots J_k)}, w_k^{(1\dots J_k)}, L_{\mathbf{Z}_k}\}$ 
end procedure

```

Fig. 8. SC-PHD SLAM per-particle update.

assumption on the map feature process imposes a high variance on its cardinality estimate, also the single-cluster process model only encodes dependencies between the map features and the vehicle, but not between map features. Recent contributions in Finite Set Statistics will allow us model SLAM in the context of general hierarchical interacting population processes, where such inter-object relationships are taken into consideration.

APPENDIX

SC-PHD SLAM PSEUDOCODE LISTINGS

This appendix contains pseudocode for our implementation of the single-cluster PHD SLAM filter. In our implementation of the filter, the joint PHD is defined by the parent particle states and their accompanying Gaussian mixture maps:

$$D_{k-1}(\mathbf{X}, \mathbf{M}) \doteq \left\{ \mathbf{X}_{k-1}^{(i)}, \mathbf{M}_{k-1}^{(i)}, \eta_{k-1}^{(i)} \right\}_{i=1}^{N_{k-1}} \quad (39)$$

$$\mathbf{M}_{k-1}^{(i)} \doteq \left\{ \mu_{k-1}^{(j)}, \mathbf{P}_{k-1}^{(j)}, w_{k-1}^{(j)} \right\}_{j=1}^{J_{k-1}^{(i)}} \quad (40)$$

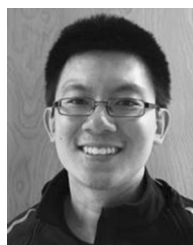
Each of the subroutines listed here accept as input and return as output a single parent particle and its corresponding map. For

numerical stability, it is recommended that weights and likelihoods be replaced with their log-equivalents, and the corresponding computations modified accordingly.

REFERENCES

- [1] R. Smith, M. Self, and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," in *Proc. UAI*, 1986, pp. 435–461.
- [2] M. Kaess, A. Ranganathan, and F. Dellaert, "iSAM: Incremental smoothing and mapping," *IEEE Trans. Robotics*, vol. 24, no. 6, pp. 1365–1378, Dec. 2008.
- [3] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "G²o: A general framework for graph optimization," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Shanghai, China, May 2011, pp. 3607–3613.
- [4] M. Cummins and P. Newman, "Appearance-only slam at large scale with fab-map 2.0," *Int. J. Robot. Res.*, vol. 30, no. 9, pp. 1100–1123, Nov. 2010.
- [5] W. P. Maddern, M. Milford, and G. Wyeth, "CAT-SLAM: Probabilistic localisation and mapping using a continuous appearance-based trajectory," *Int. J. Robot. Res.*, vol. 31, no. 4, pp. 429–451, 2012.
- [6] M. R. Walter, R. M. Eustice, and J. J. Leonard, "Exactly sparse extended information filters for feature-based slam," *Int. J. Rob. Res.* vol. 26, no. 4, pp. 335–359, Apr. 2007 [Online]. Available: <http://dx.doi.org/10.1177/02783649060075026>
- [7] L. Pedraza, D. Rodríguez-Losada, F. Matía, G. Dissanayake, and J. V. Miró, "Extending the limits of feature-based slam with b-splines," *IEEE Trans. Robot.*, vol. 25, no. 2, pp. 353–366, 2009.
- [8] A. Mallios, P. Ridao, D. Ribas, F. Maurelli, and Y. Petillot, "EKF-SLAM for AUV navigation under probabilistic sonar scan-matching," in *Proc. Intell. Robots Syst. (IROS), IEEE/RSJ Int. Conf.*, Oct. 2010, pp. 4404–4411.
- [9] J. Neira and J. Tardós, "Data association in stochastic mapping using the joint compatibility test," *IEEE Trans. Robot. Autom.*, vol. 17, no. 6, pp. 890–897, Dec. 2001.
- [10] J. Civera, O. G. Grasa, A. J. Davison, and J. M. M. Montiel, "1-point RANSAC for extended kalman filtering: Application to real-time structure from motion and visual odometry," *J. Field Robot.* vol. 27, pp. 609–631, Sep. 2010 [Online]. Available: <http://dx.doi.org/10.1002/rob.v27:5>
- [11] R. P. S. Mahler, *Statistical Multisource-Multitarget Information Fusion*. Norwood, MA, USA: Artech House, 2007.
- [12] R. Mahler, "Multitarget bayes filtering via first-order multitarget moments," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 39, no. 4, pp. 1152–1178, Oct. 2003.
- [13] R. Mahler, "PHD filters of higher order in target number," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 43, no. 4, pp. 1523–1543, Oct. 2007.
- [14] B.-T. Vo, B.-N. Vo, and A. Cantoni, "The cardinality balanced multi-target multi-bernoulli filter and its implementations," *IEEE Trans. Signal Process.*, vol. 57, no. 2, pp. 409–423, Feb. 2009.
- [15] B.-N. Vo, S. Singh, and A. Doucet, "Sequential monte carlo methods for multitarget filtering with random finite sets," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 41, no. 4, pp. 1224–1245, Oct. 2005.
- [16] D. Clark and J. Bell, "Multi-target state estimation and track continuity for the particle phd filter," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 43, no. 4, pp. 1441–1453, Oct. 2007.
- [17] B.-N. Vo and W.-K. Ma, "The gaussian mixture probability hypothesis density filter," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4091–4104, Nov. 2006.
- [18] B.-T. Vo, B.-N. Vo, and A. Cantoni, "Analytic implementations of the cardinalized probability hypothesis density filter," *IEEE Trans. Signal Process.*, vol. 57, no. 7, pp. 3553–3567, Jul. 2007.
- [19] D. Clark and J. Bell, "Convergence results for the particle PHD filter," *IEEE Trans. Signal Process.*, vol. 54, no. 7, pp. 2652–2661, Jul. 2006.
- [20] D. Clark and B. Vo, "Convergence analysis of the Gaussian mixture PHD filter," *IEEE Trans. Signal Process.*, vol. 55, no. 4, pp. 1204–1212, Apr. 2007.
- [21] A. Johansen, S. S. , D. A. , and V. B.-N. , "Convergence of the smc implementation of the phd filter," *Methodol. Comput. Appl. Probab.*, vol. 8, no. 2, pp. 265–291, Jun. 2006.
- [22] F. Caron, M. Del Moral, M. Pace, and B.-N. Vo, "On the stability and the approximation of branching distribution flows, with applications to nonlinear multiple target filtering," *J. Stochast. Anal. Applicat.*, vol. 29, no. 6, pp. 951–997, 2011.
- [23] D. Clark and J. Bell, "Bayesian multiple target tracking in forward scan sonar images using the phd filter," *IEE Proc. Radar, Sonar, Nav.*, vol. 152, no. 5, pp. 327–334, 2005.

- [24] C. Haworth, Y. de Saint-Pern, D. Clark, E. Trucco, and Y. Petillot, "Detection and tracking of multiple metallic objects in millimetre-wave images," *Int. J. Comput. Vis.*, vol. 71, no. 2, pp. 183–196, 2007.
- [25] Y. Petetin, D. Clark, B. Ristic, and D. Maltese, "A tracker based on a cphd filter approach for infrared applications," in *Proc. SPIE*, 2011, vol. 8050, p. 80500N.
- [26] D. Clark, A. Cemgil, P. Peeling, and S. Godsill, "Multi-object tracking of sinusoidal components in audio with the gaussian mixture probability hypothesis density filter," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust.*, 2007, pp. 339–342.
- [27] M. Uney, S. Julier, D. Clark, and B. Ristic, "Monte carlo realisation of a distributed multi-object fusion algorithm," in *Proc. Sensor Signal Process. for Defence Conf. (SSPD '10)*, 2010.
- [28] C. Lundquist, L. Hammarstrand, and F. Gustafsson, "Road intensity based mapping using radar measurements with a probability hypothesis density filter," *IEEE Trans. Signal Process.*, vol. 59, no. 4, pp. 1397–1408, Apr. 2011.
- [29] M. Cannaud, L. Mihaylova, N.-E. E. Faouzi, R. Billot, and J. Sau, "A probabilistic hypothesis density filter for traffic flow estimation in the presence of clutter," in *Proc. IEEE Sensor Data Fusion Workshop: Trends, Solut., Applicat.*, 2012, pp. 31–36.
- [30] T. M. Wood, C. A. Yates, D. A. Wilkinson, and G. Rosser, "Simplified multitarget tracking using the phd filter for microscopic video data," *IEEE Trans. Circuits Syst. Video Technol.* vol. 22, no. 5, pp. 702–713, May 2012.
- [31] R. Hoseinnezhad, B.-N. Vo, B.-T. Vo, and D. Suter, "Visual tracking of numerous targets via multi-bernoulli filtering of image data," *Pattern Recogn.* vol. 45, no. 10, pp. 3625–3635, Oct. 2012 [Online]. Available: <http://dx.doi.org/10.1016/j.patcog.2012.04.004>
- [32] J. Mullane, B.-N. Vo, M. D. Adams, and B.-T. Vo, "A random-finite-set approach to Bayesian SLAM," *IEEE Trans. Robot.*, vol. 27, no. 2, pp. 268–282, Apr. 2011.
- [33] J. Mullane, B.-N. Vo, M. Adams, and B.-T. Vo, *Random Finite Sets for Robot Mapping and SLAM—New Concepts in Autonomous Robotic Map Representations*, ser. Springer Tracts in Advanced Robotics. New York, NY, USA: Springer, 2011, vol. 72.
- [34] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A factored solution to the simultaneous localization and mapping problem," in *Proc. AAAI Nat. Conf. Artif. Intell.*, Edmonton, AB, Canada, 2002, AAAI.
- [35] D. Moratuwake, B.-N. Vo, and D. Wang, "A hierarchical approach to the multi-vehicle SLAM problem," in *Proc. Int. Conf. Inf. Fusion*, Jul. 2012.
- [36] A. Swain and D. Clark, "First-moment filters for spatial independent cluster processes," in *Proc. SPIE*, 2010, vol. 7697, p. 76970I.
- [37] A. Swain and D. Clark, "Extended object filtering using spatial independent cluster processes," in *Proc. 13th Inf. Fusion Conf. (FUSION)*, 2010, pp. 1–8.
- [38] D. E. Clark and J. Houssineau, "Faa di Bruno's formula for Gateaux differentials and interacting stochastic population processes," *ArXiv e-prints*, Feb. 2012.
- [39] B. Ristic and D. E. Clark, "Particle filter for joint estimation of multi-object dynamic state and multi-sensor bias," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2012.
- [40] B. Ristic, D. E. Clark, and N. Gordon, "Calibration of tracking systems using detections from non-cooperative targets," in *Proc. IEEE Sens. Data Fusion Workshop: Trends, Solut., Applicat.*, 2012.
- [41] C.-C. Wang, C. Thorpe, and S. Thrun, "Online simultaneous localization and mapping with detection and tracking of moving objects: Theory and results from a ground vehicle in crowded urban areas," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2003, pp. 842–849.
- [42] Y.-T. Wang, C.-H. Sun, and M.-J. Chiou, "Detection of moving objects in image plane for robot navigation using monocular vision," *EURASIP J. Adv. Signal Process.*, vol. 2012, p. 29, 2012.
- [43] D. J. Daley and D. Vere-Jones, *An Introduction to the Theory of Point Processes, Vol. 1*, ser. Probability and its Applications (New York), 2nd ed. New York, NY, USA: Springer-Verlag, 2003, elementary theory and methods.
- [44] A. Swain and D. Clark, "The single-group phd filter: An analytic solution," in *Proc. Int. Conf. Inf. Fusion*, 2011.
- [45] A. Swain and D. Clark, "The single-group phd filter: An analytic solution," in *Proc. Int. Conf. Inf. Fusion*, 2011.
- [46] R. Mahler, B.-T. Vo, and B.-N. Vo, "Cphd filtering with unknown clutter rate and detection profile," *IEEE Trans. Signal Process.*, vol. 59, no. 8, pp. 3497–3513, Aug. 2011.
- [47] J. Houssineau and D. Laneuville, "Phd filter with diffuse spatial prior on the birth process with applications to GM-PHD filter," in *Proc. 13th Conf. Inf. Fusion (FUSION)*, Jul. 2010, pp. 1–8.
- [48] D. Salmund, "Mixture reduction algorithms for point and extended object tracking in clutter," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 45, no. 2, pp. 667–686, Apr. 2009.
- [49] D. Schieferdecker and M. Huber, "Gaussian mixture reduction via clustering," in *Proc. 12th Int. Conf. FUSION '09*, 2009.
- [50] J. E. Guivant and E. M. Nebot, "Optimization of the simultaneous localization and map-building algorithm for real-time implementation," *IEEE Trans. Robot.*, vol. 17, no. 3, pp. 242–257, Jun. 2001.
- [51] E. M. Nebot, Ute Data Parameters, [Online]. Available: http://www-personal.acfr.usyd.edu.au/nebot/experimental_data/modeling_info/Ute_modeling_info.htm Feb. 2003
- [52] D. Schuhmacher, B.-T. Vo, and B.-N. Vo, "A consistent metric for performance evaluation of multi-object filters," *IEEE Trans. Signal Process.*, vol. 56, no. 8, pt. 1, pp. 3447–3457, Aug. 2008.
- [53] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, Feb. 2002.



Chee Sing Lee is a Ph.D. candidate at the University of Girona. His current research is centered on simultaneous localization and mapping, and autonomous underwater vehicles. He also has prior experience working in computer vision, machine learning, FPGA design, and electric vehicles. He graduated from Oregon State University with an H.B.S. in Electrical and Electronic Engineering in 2008, and earned an Erasmus Mundus Masters in Computer Science and Robotics (VIBOT) in 2010.



Daniel E. Clark is a Reader at Heriot-Watt University and currently holds the Royal Academy of Engineering/EPSRC Research Fellowship. His research interests are in the development of the theory and applications of multi-object estimation algorithms for sensor fusion problems. He has collaborated closely with government and industry internationally on a number of projects spanning theoretical algorithm development to practical deployment. He has held positions of Visiting Researcher at the University of Melbourne, Visiting

Professor at the University of Western Australia, and Visiting Scientist at Thales Netherlands. He obtained his Ph.D. at Heriot-Watt in 2006, Diploma in Computer Science from the University of Cambridge in 2002, and a B.Sc. (Hons) in Mathematics from the University of Edinburgh. Daniel Clark is a Royal Academy of Engineering/EPSRC Research Fellow.



Joaquim Salvi graduated in Computer Science at the Technical University of Catalonia in 1993, received the DEA (MSc) in Computer Science in July 1996 and the Ph.D. in Industrial Engineering in 1998 both at the University of Girona, Spain. He is Professor of Computer Vision at the Computer Architecture and Technology Department and at the Computer Vision and Robotics Group, University of Girona; and he was a visiting professor at the Ocean Systems Lab, Heriot-Watt University (UK). He is involved in some governmental projects and technology transfer

contracts to industry. His current interests are in the field of computer vision and mobile robotics, focused on visual SLAM, structured light, stereovision, and camera calibration. He is the leader of the 3D Perception Lab and charter member of the spinoffs AQSense and BonesNotes. Dr. Salvi received the Best Thesis Award in engineering for his Ph.D.