Extended Object Tracking: Introduction, Overview and Applications

Karl Granström, Marcus Baum

Abstract—This article provides an elaborate overview of current research in extended object tracking. We provide a clear definition of an extended object and discuss its delimitation to other object types and sensor models. Next, different shape models and possibilities to model the number of measurements are extensively discussed. Subsequently, we give a tutorial introduction to two basic and well used extended object tracking methods – the random matrix approach and random hyper-surface approach. The next part treats approaches for tracking multiple extended objects and elaborates how the large number of feasible association hypotheses can be tackled using both Random Finite Set (RFS) and Non-RFS multi-object trackers. The article concludes with a summary of current applications, where three example applications involving LIDAR, RGB, and RGB-D sensors are highlighted.

I. INTRODUCTION

Multiple Target Tracking (MTT) denotes the process of successively determining the number and states of multiple dynamic objects based on noisy sensor measurements. Nowadays, tracking systems are a key technology for many technical applications in areas such as robotics, surveillance, autonomous driving, automation, medicine, and sensor networks.

Traditionally, MTT algorithms have been tailored for scenarios with multiple remote objects that are far away from the sensor and well separated from each other, e.g., as in radar-based air surveillance. In such scenarios, a object is not always detected by the sensor, and if it is detected, at most one sensor resolution cell is occupied by the object. From traditional scenarios, specific assumptions on the mathematical model of MTT problems have evolved including the so-called “small object” assumptions:

- The objects evolve independently,
- each object can be modeled as a point without any spatial extent, and
- each object gives rise to at most a single measurement per time frame/scan.

MTT based on the “small object” assumptions is a highly complex problem. In addition to noise, missed measurements, and clutter, the number of objects is unknown and usually time-varying. The most common approaches to MTT are: (i) Multiple Hypothesis Tracking (MHT) [16] and Joint Probabilistic Data Association (JPDA) [3], (ii) Probabilistic Multiple Hypothesis Tracking (PMHT) [120], [135], and (iii) Random Finite Sets (RFS) [83], [85]. The MHT type approaches involve propagating object track hypotheses in time and calculating their likelihoods, the JPDA type approaches blend data association probabilities on a scan-by-scan basis. The PMHT approach allows multiple assignments to the same object, which results in an efficient method using the Expectation-Maximization (EM) framework. The RFS type approaches rely on modeling the objects and the measurements as random sets.

Today, there is still a huge variety of applications for which the “small object” assumptions are reasonable. However, due to rapid advances in sensor technology in the recent years, it is becoming increasingly common that objects occupy several sensor resolution cells. Furthermore, novel applications with objects in the near-field of sensors, e.g., in mobile robotics and autonomous driving, often render the “small object” assumptions invalid.

The tracking of an object that might occupy more than one sensor cell leads to the so-called extended object tracking or extended target tracking problem. An extended object gives rise to a varying number of potentially noisy measurements from different spatially distributed measurement sources, also referred to as reflection points. The shape of the object is usually unknown and can even vary over time, and the objective is to recursively determine the shape of the object plus its kinematic parameters. Due to the nonlinearity of the resulting estimation problem, already tracking a single extended object is in general a highly complex problem for which elaborate nonlinear estimation techniques are required.

Although often misunderstood – extended object tracking, as defined above, is fundamentally different from typical contour tracking problems in computer vision [139]. In vision-based contour tracking [139], a complete (RGB) image is available at each time frame and one extracts a contour from each image that is tracked over time. In extended object tracking, one works with a few (usually two or three-dimensional) measurements per time step, i.e., a sparse point cloud. It is nearly always impossible to extract a shape only based on the measurement from one time instant. The object shape can only be determined if measurements over several time steps are systematically accumulated and fused under incorporation of the (unknown) object motion and sensor noise. An illustration of the difference between point object tracking, extended object tracking, and contour tracking is given in Figure 1 based on the experiments from [11].

In many practical applications it is necessary to track multiple extended objects, where no measurement-to-object associations are available. Unfortunately, data association be-
The objective of this article is to
(i) provide an elaborate up-to-date introduction to the extended object tracking problem,
(ii) introduce basic concepts, models, and methods for shape estimation of a single extended object,
(iii) introduce the basic concepts, models, and methods for tracking multiple extended objects,
(iv) point out recent applications and future trends.

Historically, the first works on extended object tracking can be traced back to [27], [28]. Already in 2004, [131] gave a short literature overview of cluster (group) tracking and extended object tracking problems. However, since then, huge progress has been made in both shape estimation of a single object and multi-(extended)-object tracking. An overview of Sequential Monte Carlo (SMC) methods for group and extended object tracking can be found in [90]. The focus of [90] lies on group object tracking and SMC methods. Hence, the content of [90] is orthogonal to this article, and the two articles complement each other. A comparison of early versions of the random matrix and random hypersurface approach was performed in [13]. Since the publication of [13], both methods have been significantly further developed.

The rest of the paper is organized as follows. In the next section some definitions are introduced, and modeling of object shape, number of measurements, and object dynamics is overviewed. Section III introduces two popular approaches to extent modeling and estimation: the random matrix model, Section III-A, and the random hypersurface model, Section III-B. Multiple extended object tracking is overviewed in Section IV and in Section V three applications are presented: tracking cars using a LIDAR, tracking groups of pedestrians using a camera, and tracking complex shapes using a RGB-D sensor. The paper is concluded in Section VI.

II. DEFINITIONS AND EXTENDED OBJECT MODELING

In this section we will first give a definition of an extended object and some related object types. Second we overview the modeling of the object shape, the object dynamics, and the number of sensor measurements per object. In comparison to modeling the object’s shape, modeling the dynamics and the number of measurements is simpler. However, it is worth while to consider all three topics.

A. Definitions

The following are definitions of object types that are relevant to this article.

- **Point object:** An object that can result in at most a single measurement per time step.
- **Extended object:** An object that can result in multiple \((\geq 1)\) spatially distributed measurements per time step.
- **Group object:** A group of two or more objects that share some common motion, and are not tracked individually but are instead treated as a single entity.
- **Multi-path object:** An object that can result in more than one measurement per time step, where the measurements are not spatially structured.

The focus of the article lies on extended objects. However, we note that it is possible – and quite common – to employ extended object tracking methods to track the shape of a group object, see, e.g., [90]. It is easy to see that extended object and group object are two very similar concepts, however some distinctions can be made that warrant two definitions instead of just one.

An extended object is a single entity, e.g., a car, an airplane, a human, or an animal. Often the shape can be assumed to be a rigid body\(^1\), however extended objects with deformable extents is also possible. A group object is a collection of objects that share some common dynamics, while still allowing

\(^1\)With the exception of the orientation of the extent, the size and shape of the object does not change over time
for individual dynamics within the group. For example, in a group of pedestrians, there is an overall group motion, but the individual pedestrians may also shift their positions within the group.

Technically, an extended object occurs when the sensor resolution is high enough so that the object will occupy more than one resolution cell. As a consequence there may be more than one measurement in different cells. In comparison, the size of a point object is such that it occupies at most one resolution cell, so that there will be at most a single measurement per time step.

In some literature, an extended object is understood to be an object that has a spatial extent, in comparison to a point object that does not have a spatial extent. The spatial extent then causes the possibility of multiple measurements, instead of just the single one. These two definitions are somewhat unfortunate though, because they are a poor representation of the underlying reality: with the exception of, e.g., certain sub-atomic particles, objects have spatial extents.

The measurements from an extended object are caused by measurement sources, which may have different meaning depending on the sensor that is used and the types of objects that are tracked. In some cases, e.g., see [13], [66], one can model a finite number of measurement sources, while in other cases it is better to model an infinite number of sources. For example, in [66] automotive radar is used to track cars, and the measurements are located around the wheelhouses of the tracked cars, i.e. there are four measurement sources. In [114] LIDAR is used to track cars, and the measurements are then located on the chassi of the car. This can be interpreted as an infinite number of points that may act as measurement sources.

Note that certain sensors measure the object’s cross-range and down-range extents (or similar object features), allowing for the extent of the object to be estimated, see e.g., [1], [12], [144]. However, by the definitions used here, this is not extended object tracking unless there are multiple such measurements.

Lastly, multi-path objects occur in scenarios that are susceptible to the multi-path phenomenon, for example when data from over-the-horizon radar is used, see, e.g., [65], [113], [123]. The difference between extended object and multi-path object lies in the distribution of the measurements: for multi-path objects a spatial distribution cannot necessarily be assumed.

B. Shape Models for Extended Objects

The types of extended object spatial distributions can be divided into two classes:

- Measurements along the boundary of the object’s extent. For measurements in 2D, this means that the measurements are noisy points on a curve. For measurements in 3D, the measurements are noisy points on either a curve or a surface. Measurements along the boundary are obtained, e.g., when LIDAR is used in automotive applications.
- Measurements inside the volume/area of the object’s extent, i.e., the measurements form a cluster. Such measurements are obtained, e.g., when radar is used to track marine vessels.

In Table I some references are listed according to the shape dimension and measurement type, and Figure 2 provides an illustration. To our knowledge there is no explicit work about the estimation of 3D shapes in 3D space, probably because there are rarely sensors for this case. However, most algorithms for 2D shapes in 2D space can be generalized rather easily to the 3D case.

When the measurements lie on the boundary of the extended object, the resulting theoretical problem shares similarities with traditional curve fitting, where a curve is to be matched with noise points [20], [39]. However, the curve fitting problem only considers static scenarios, i.e., non-moving curves. Additionally, the noise is usually isotropic and non-recursive non-Bayesian methods have been developed. Hence, curve fitting algorithms usually cannot directly be applied in the extended object tracking context. For a discussion of the rare Kalman filter-based approaches for curve fitting, we refer to [103], [143].

It is also useful to distinguish different complexity levels for describing the shape of the object. Different shape complexities might require different approximations and algorithms. The different ways to model this type of extended object tracking scenario are here divided into three complexity levels:

1) The simplest level of modeling is to not model the shape at all, i.e. to only estimate the object’s kinematic properties. This approach has lowest computational complexity and the flexibility to track different type of objects is high because this model, even though it is simplistic in terms of object shape, is often applicable (with varying degree of accuracy).

2) A more advanced level of modeling is to assume a specific basic geometric shape for the object, such as

<table>
<thead>
<tr>
<th>TABLE I</th>
<th>SHAPE DIMENSIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Curve in 2D/3D space:</td>
<td>[4], [12], [46], [48], [103], [143]</td>
</tr>
<tr>
<td>Surface in 2D space:</td>
<td>[7], [21], [80], [71], [23], [109], [105]</td>
</tr>
<tr>
<td>Surface in 3D space:</td>
<td>[107], [115], [148]</td>
</tr>
</tbody>
</table>

Fig. 2. One-dimensional (a) and two-dimensional (b) shape.

Fig. 3. Shape complexities.
an ellipse, a line or a rectangle.

3) The most advanced approach is to construct a measurement model that is capable of handling a broad variety of both different shapes and different measurement appearances. While such a model would be most general, it could also prove to be overly computationally complex.

The three complexity levels are illustrated in Figure 3 and some references whose shape modelling fall into the latter two categories are listed in Table II.

What the correct choice of complexity level is, is a question that does not have a simple answer. In general, the more complex the shape, the more measurements (with less noise) are required to get a reasonable shape estimate. Furthermore, it depends on the type of sensor that is used, the types of objects, their motions, and what the tracking output will be used for. In some scenarios it may be sufficient to know the position of each object, in other scenarios it is necessary to have a detailed estimate of the size and shape of each object.

For example, in [48] it is shown that using LiDAR data bicycles can be tracked fairly accurately without modeling the extent, however if the shape of a bicycle is approximated by a stick with unknown length, estimation performance is improved. Specifically, by modeling the shape it becomes possible to capture rotations of the shape, and thus capture the onset of turning maneuvers. Without a shape estimate, the turning is captured at a later time.

The stick shape is a suitable approximation of the shape of a bicycle [48] in 2D, whereas a rectangle can be used to approximate the shape of a car in 2D [50], [61], [101]. With a Gaussian process approach it is possible to estimate a general object shape [130]. For a car in 2D this allows for rounded corners rather than sharp ones, which is a more accurate model of actual cars than a rectangle is. In terms of estimating an arbitrary object shape, the literature contains at least two different types of approach: either the shape is modeled as a curve with some parametrization [6], [80], [130], or the shape is modeled as combination of ellipses [62], [74], [75].

C. Modeling the number of measurements

The possibility of multiple measurements per object imply the need for a probabilistic model of the number of measurements per object. The expected number of measurements contains less information about the object compared to position, kinematics, or extent. However, multiple object tracking studies have shown that when multiple objects of equal size and shape are spatially very close to each other, tracking performance can deteriorate significantly if the expected number of measurements is unknown, see, e.g., [51]. In the literature two alternatives for modeling the number of measurements can be found.

The first model, proposed in [44], [46], models the number of measurements as Poisson distributed, with a Poisson rate $\gamma$ that is a function of the object’s state. More specifically, the extended object measurements are modeled as a Poisson point process. This model can be understood to imply that the extended object is sufficiently far away from the sensor for the measurements to resemble a cluster of points, rather than a structured ensemble. However, the model has been used successfully in multiple object scenarios where the object measurements show a high degree of structure, see, e.g., [48], [55], [61].

The conjugate prior for an unknown Poisson rate is the gamma distribution, see, e.g., [43]. By augmenting the estimated distribution for the kinematic state and the extent state with a gamma distribution for the Poisson rate, an individual Poisson rate can be estimated for each extended object, see [56].

The second model, proposed in [108], [109], models the number of measurements as multi-Bernoulli distributed. This model can be understood to imply that the extended object has some number, $L$ say, of reflection points. The reflection points are detected independently of each other, and the $\ell$th reflection point has measurement probability $p_D$. In technical terms, the measurement process for each reflection point can be described as a Bernoulli RFS [83], [85], and the measurement process for the extended object is a multi-Bernoulli RFS [83], [85].

If the measurement probabilities are equal for all reflection points, $p_D = p_D, \forall \ell$, then the number of measurements is Binomial distributed with parameters $L$ and $p_D$. For known $L$, the conjugate prior for an unknown $p_D$ is the Beta distribution.

Bayesian approaches to estimating unknown $L$ given a known $p_D$, or estimating both $L$ and $p_D$, have to the best of our knowledge not been presented. However, a simple heuristic for determining $L$, under the assumption that $p_D$ is known, is given in [108].

Note that the multi-Bernoulli model has two degrees of freedom (two parameters), whereas the Poisson model has only one. Thus, with the multi-Bernoulli model it is possible to determine both mean and variance, while the Poisson model’s variance is equal to the mean by definition.

D. Modeling the dynamics

The object dynamic model describe how the object state evolves over time; for a moving object this describes how the object moves. This involves the position and any states that describe the motion—e.g., velocity, acceleration, turnrate—however, it also involves descriptions of how the extent changes over time (typically it rotates when the object turns) and how the number of measurements changes over time (often there are more measurements the closer to the sensor the object is).

In many cases extended object dynamics can be modeled using any of the of the models that are standard in point object tracking, see [10] for a comprehensive overview. When there

<table>
<thead>
<tr>
<th>OBJECT SHAPE</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stick</td>
<td>[4], [17], [46], [48], [125]</td>
</tr>
<tr>
<td>Circle</td>
<td>[6], [109]</td>
</tr>
<tr>
<td>Ellipse</td>
<td>[6], [24], [80], [74], [75], [105], [130]</td>
</tr>
<tr>
<td>Rectangle</td>
<td>[50], [61]</td>
</tr>
<tr>
<td>Arbitrary shape</td>
<td>[6], [62], [74], [125], [80], [130]</td>
</tr>
</tbody>
</table>
are multiple objects present a common assumption is that the objects evolve independently of each other, resulting in the object estimates being predicted independently. In point object tracking, where several objects form groups while remaining distinguishable, it is possible to apply, e.g., leader-follower models, allowing for the individual objects to be predicted dependently, see e.g. [21], [98].

III. SHAPE ESTIMATION METHODS

This section discusses two basic deterministic methods for estimating the shape of a single extended object, i.e., the random matrix approach and the random hypersurface approach. These two methods are deterministic in the sense that no random sampling is performed at any time. For a detailed discussion of Monte Carlo methods, we refer to [90].

A. Random matrix model

In this section we overview an extended object model that is known as the random matrix model. This model assumes that the object shape can be approximated by an ellipse, and that the measurements are Gaussian distributed around that object’s centroid (center of mass). The ellipse shape may seem limiting, however the model is applicable to many real scenarios, e.g. pedestrian tracking using LIDAR [55].

1) Original approach: The random matrix model was originally proposed in [71], and it models the extended object state as the combination of a kinematic state vector \( x_k \) and an extent matrix \( X_k \). The vector \( x_k \) represents the object’s position and kinematics, and the matrix \( X_k \) represents the object’s size and shape, i.e. its spatial extent. The matrix \( X_k \) is modeled as being symmetric and positive definite, which implies that the object shape is approximated by an ellipse.

At each time step there is a set of \( n_k \) independent Cartesian measurements, denoted \( \mathbf{Z}_k = \{ \mathbf{z}_k \}^{n_k}_{j=1} \). The measurements are assumed independent, and conditioned on the number of measurements \( n_k \) and the object state \( x_k, X_k \) the measurement set likelihood is

\[
p(\mathbf{Z}_k|n_k, x_k, X_k) = \prod_{j=1}^{n_k} p(\mathbf{z}_k^j|x_k, X_k) \quad (1)
\]

The object generated measurements are modeled as Gaussian distributed,

\[
p(\mathbf{z}_k|x_k, X_k) = \mathcal{N}(\mathbf{z}_k; \mathbf{H}x_k, X_k) . \quad (2a)
\]

where the noise covariance matrix is the extent matrix, and \( \mathbf{H} \) is a measurement model that picks out the Cartesian position from the kinematic vector \( x_k \).

For Gaussian measurements, the conjugate priors for unknown mean and covariance are the Gaussian and the inverse Wishart distributions, respectively. This motivates the object state distribution [71]

\[
p(x_k, X_k|\mathbf{Z}^k) = p(x_k|X_k, \mathbf{Z}^k) p(X_k|\mathbf{Z}^k) = \mathcal{N}(x_k; m_k[k], P_k[k] \otimes X_k) \times IW_d(X_k; v_k[k], V_k[k]) , \quad (2b)
\]

where the extent matrix is inverse Wishart distributed, and the kinematic vector, conditioned on the extent, is Gaussian distributed. In this model the kinematic state \( x_k \) consists of a spatial state component \( r_k \) (object position), and derivatives of \( r_k \) (typically velocity and acceleration, although higher derivatives are possible) [71]. Non-linear dynamics, such as turn-rate, are not included in the kinematic vector. The measurement update is linear [71], and a derivation of the predicted likelihood can be found in [55, Appendix A].

2) Improved noise modeling: An implicit assumption of the original random matrix model [2] is that the measurement noise is negligible in size compared to the size of the extent. In some scenarios this assumption does not hold, and will lead to a biased estimate.

To alleviate this problem each measurement \( z_k \) can thought of as a noisy measurement of a reflection point \( y_k \) that is located somewhere on the extended object. Each reflection point is modeled as a point randomly sampled from the object’s extension. The measurement likelihood becomes

\[
p(\mathbf{z}_k|x_k, X_k) = \int p(\mathbf{z}_k|y_k, x_k, X_k) p(y_k|x_k, X_k) dy_k \quad (3a)
\]

\[
= \int p(\mathbf{z}_k, y_k|x_k, X_k) dy_k \quad (3b)
\]

In other words, the measurement likelihood [3] is the marginalization of the reflection point \( y \) out of the estimation problem.

In many scenarios the measurement noise can be modeled as zero mean Gaussian,

\[
p(\mathbf{z}_k|y_k, x_k, X_k) = \mathcal{N}(\mathbf{z}_k; y_k, R(y_k)) \quad (4)
\]

where \( R(y) \) is the noise covariance matrix that possibly is a function of the reflection point. One choice is to model the reflection points as uniform samples from the object shape,

\[
p(y_k|x_k, X_k) = \mathcal{U}(y_k; x_k, X_k) . \quad (5)
\]

The drawback of this choice is that the integral \( 3 \) is not analytically tractable even for constant noise covariance \( R(y) = R \).

As shown by Feldmann et al [38], for an elliptically shaped object the uniform distribution [5] can be approximated by a Gaussian distribution

\[
p(y_k|x_k, X_k) = \mathcal{N}(y_k; \mathbf{H}x_k, z X_k) \quad (6)
\]

where \( z \) is a scaling factor. A simulation study in [38] showed that \( z = 1/4 \) is a good parameter setting. This is illustrated in Figure 4 see subfigures a and b.

Together Gaussian noise model [4] and the Gaussian approximation [6] give the following likelihood

\[
p(\mathbf{z}_k|x_k, X_k) = \int \mathcal{N}(\mathbf{z}_k; y_k, R(y_k)) \mathcal{N}(y_k; \mathbf{H}x_k, z X_k) dy_k \quad (7)
\]

If the noise matrix \( R(y_k) \) is constant, the integral results in the measurement likelihood suggested by Feldmann et al [36–38],

\[
p(\mathbf{z}_k|x_k, X_k) = \mathcal{N}(\mathbf{z}_k; H_k x_k, z X_k + R) , \quad (8a)
\]
An example with elliptic extent $X$ and circular measurement noise covariance $R$ is given in Figure 4, see subfigure c. The object state density is modeled as 

$$p(x_k|X_k|Z^k) = p(x_k|Z^k) p(X_k|Z^k)$$

(8b)

$$= N(x_k; m_{k|k}', P_{k|k}') \times IW_d(X_k; v_{k|k}', V_{k|k}')$$

(8c)

Note the assumed independence between the kinematic state $x_k$ and $X_k$ in (8b), an assumption that cannot be fully theoretically justified.

Despite this theoretical drawback, there are some practical advantages to using the state distribution (8b), instead of (2c). This model allows for a more general class of kinematic state vectors $x_k$, e.g. including non-linear dynamics such as heading and turn-rate, and the Gaussian covariance is no longer intertwined with the extent matrix. Further, the measurement model is better when the size of the extent and the size of the sensor noise are within the same order of magnitude [38].

The assumed independence between $x_k$ and $X_k$ is alleviated in practice by the measurement update which provides for the necessary interdependence between kinematics and extent estimation, see [38].

Using the measurement likelihood (8a) the extent estimate is unbiased, however the measurement update is no longer linear and must be approximated. Some alternative non-linear updates have been proposed. Originally in [38] the extent matrix was assumed to be approximately equal to the predicted estimate $X_k \approx \hat{X}_{k|k-1} = E[X_k|Z^{k-1}]$, and using Cholesky factorization an update was derived.

An update based on variational Bayesian approximation is presented in [94]; a simulation study shows that the variational update has smaller estimation error than the update based on Cholesky factorization [38], at the price of higher computational cost. An update based on linearization of the natural logarithm of the measurement likelihood (8a) is presented in [2]. A simulation study shows that the log-linearized update gives results that almost match the variational update, at a lower computational cost.

In scenarios where radar sensors are used the noise matrix $R(y_k)$ cannot be assumed constant, because of the polar measurement noise, see e.g. [126]. After conversion to Cartesian coordinates the spread of the measurements is larger the further the object is from the sensor, see Figure 4, subfigures d and e. This can be handled by approximating the reflection point by the predicted object position $y_k \approx H_k \hat{x}_{k|k-1} = H_k E[X_k|Z^{k-1}]$ [126].

A third measurement model is [73],

$$p(z_k|x_k, X_k) = N(z_k; H_k x_k, B_k X_k B_k^T)$$

(9)

where $B_k$ is a known parameter matrix. Note that this measurement model was designed for the dependent extended object state [23]. Under the assumption $X_k \approx \hat{X}_{k|k-1} = E[X_k|Z^{k-1}]$ the model (9) incorporates (8a) approximately when $B_k = (z \hat{X}_{k|k-1} + R)^{1/2} \hat{X}_{k|k-1}^{-1/2}$.

3) Dynamic modeling: In the original random matrix model [71] the transition density is modeled as

$$p(x_{k+1}, X_{k+1}|x_k, X_k) \approx p(x_{k+1}|x_k, X_k) p(X_{k+1}|X_k),$$

(10)

and in [38] a slightly different transition density was proposed,

$$p(x_{k+1}, X_{k+1}|x_k, X_k) \approx p(x_{k+1}|x_k) p(X_{k+1}|X_k).$$

(11)

Both works [38], [71] use a linear Gaussian transition density for the kinematic vector, and for the extent a simple heuristic is used in which the expected value is kept constant and the variance is increased [71].

This model for the extent’s time evolution is sufficient when the object manoeuvres are sufficiently slow. In practice, this means as the object turns slowly enough for the rotation of the extent to be very small from one time step to another. The kinematics transition density $p(x_{k+1}|x_k)$ in (11) is assumed independent of the extent. This neglects factors such as wind resistance, which can be modeled as a function of the extent $x_k$, however the assumption is necessary to retain the functional form (8b) in a Bayesian recursion.

An alternative to the heuristic extent predictions from [38], [71] is to use a Wishart transition density [60], [71], [73], [78].

$$p(X_{k+1}|X_k) = W_d(X_{k+1}; n_k, X_k/n_k).$$

(12)

Here the expected value is constant, and the parameter $n_k > 0$ governs the noise level of the prediction: the smaller $n_k$ is, the higher the process noise.

In [73] transformations of the extent are allowed via known parameter matrices $A_k$,

$$p(X_{k+1}|X_k) = W_d(X_{k+1}; \delta_k, A_k X_k A_k^T).$$

(13)

The parameter matrices correspond to, e.g., rotation matrices.
The extent transition density \( p(X_{k+1} | X_k) \) in (10), (11), (12), and (13) assumes independence of the prior kinematic state \( x_k \). The extent of a object going through a turning manoeuvre will typically rotate during the turn, because the extent is aligned with the object's heading. This implies that the extent transition density should be dependent on the turn-rate, i.e., it should be dependent on the kinematic state \( x_k \).

The inverse Wishart transition density is generalized in [54], [60] to allow for transformation matrices \( M(x_k) \) that are functions of the kinematic state,

\[
p(X_{k+1} | x_k, X_k) = \mathcal{W}_d \left( X_{k+1} ; n_k, \frac{M(x_k)X_kM(x_k)^T}{n_k} \right),
\]

which means that the rotation angle can be coupled to, e.g., the turn-rate and estimated online. A comparison of the models (11), (13) and (14) is given in [60], where (14) is shown to give lowest estimation and prediction errors.

When there are many measurements per object the measurement update will dominate the prediction and compensate for dynamic motion modeling errors. However, when multiple objects are located next to each other the prediction is important even in scenarios with many measurements per object, especially when two or more object a located next to each other [60], [64].

4) Further extensions of the random matrix model: The random matrix models have been integrated into several different MTT algorithms, e.g., the Probabilistic Multi-Hypothesis Tracking (PMHT) framework [119], see [132]–[134]. The model has also been used in PHD- and CPHD-filters for multiple extended object tracking in clutter, see [55], [81]. In many of the MTT algorithms it is necessary to maintain several object hypotheses; the number of hypotheses can be reduced using the merging algorithm presented in [55]. Elliptically shaped group objects are tracked under kinematical constraints in [70]. The radar doppler rate is integrated into the measurement modeling in [175]. New object spawning, and merging of two object's into a single object, is modeled within the random matrix framework in [59].

Extended objects with irregular shapes can be tracked by approximating the shape as a combination of several elliptically shaped subobjects. Using multiple instances of a simpler shape alleviates the limitations posed by the implied elliptic object shape and also retains, on a subobject level, the relative simplicity of the random matrix model. In [75] a single extended object model is given where the extended object is a combination of multiple subobjects with kinematic state vectors \( \mathbf{x}_k^{(i)} \) and extent matrices \( X_k^{(i)} \). Each subobject is modeled using (22).

\[
\mathcal{N} \left( \mathbf{x}_k^{(i)} ; m_k^{(i)}, P_k^{(i)} \otimes X_k^{(i)} \right) \mathcal{W}_d \left( X_k^{(i)} ; v_k^{(i)}, V_k^{(i)} \right).
\]

Note that this model assumes independence between the subobjects. By modeling the subobject kinematic vectors as dependent random variables estimation performance can be improved significantly, see [62], [64].

As the number of ellipses grows, their combination can form nearly any given shape.

B. Random Hypersurface Approach

The random hypersurface approach [5], [7] constitutes a general extended object tracking framework that employs

- a parametric representation of the shape contour,
- a Gaussian distribution for representing the uncertainty of the joint state vector of the kinematic and shape parameters, and
- nonlinear Kalman filters for performing the measurement update.

In contrast to the random matrix model that inherently relies on the ellptic shape, the random hypersurface model can be designed for general star-convex shapes (without using multiple sub-objects). However, the increased flexibility comes at the price of much more complex closed-form formulas.

Finally, an overview of recent developments and trends in the context of random hypersurface models is given.

1) Review – Nonlinear Kalman Filtering: Consider a general nonlinear measurement function (time index is omitted) in the form

\[
z = h(x, v), \]

which maps the state \( x \) and the noise \( v \) to the measurement \( z \).

We assume that both the prior probability density function of the state and noise density are Gaussian, i.e., \( p(x) = \mathcal{N}(x; m, P) \) and \( p(v) = \mathcal{N}(v; 0, V) \). In order to calculate the posterior density function

\[
p(x | z) = \frac{p(z | x) \cdot p(x)}{p(z)},
\]

it is necessary to determine the likelihood function \( p(z | x) \) based on (16). Unfortunately, as the noise in (16) is non-additive, no general closed-form solution for the likelihood is available. As a consequence, nonlinear estimators that work with the likelihood function (e.g., standard particle filters) cannot be applied directly to this kind of measurement equation. However, there are nonlinear filters that do not explicitly calculate the likelihood function – instead they exclusively work with the measurement equation (16). The most prominent examples are nonlinear Kalman filters, which directly apply the Kalman filter formulas to the nonlinear measurement equation (16) in order to approximate the mean \( m^+ \) and covariance \( P^+ \) of the posterior density (17) as

\[
m^+ = m + \text{Cov}[z, x] P^{-1} (z - E[z])
\]

\[
P^+ = P - \text{Cov}[x, z] \text{Cov}[z, z]^{-1} \text{Cov}[z, x].
\]

Of course, in case of high nonlinearity of the measurement equation, this can be a pretty rough approximation. The exact posterior is only obtained in case of a linear measurement equation.

Analytic expression for the required moments \( E[z] \), \( \text{Cov}[z, x] \), and \( \text{Cov}[z, z] \) in (18) and (19) are only available for special cases, e.g., polynomial measurement equations. However, a huge variety of approximate methods has been
developed in the past such as the unscented transform [68]. More advanced methods are discussed in detail in [76].

2) Random Hypersurface Model for Star-Convex Shapes:
In the following, it is shown how the extended object tracking problem can be formulated as a measurement equation with non-additive noise [16] using the concept of a random hypersurface model. Based on the derived measurement equation, nonlinear Kalman filters can be used to estimate the shape of extended objects as described above.

For this purpose, we first define a suitable parameterization of a star-convex shape based on the so-called radius function \( r(p_k, \phi) \), which maps a shape parameter vector \( p_k \) and an angle \( \phi \) to a contour point (relative to a center \( d_k \)). A reasonable (finite dimensional) shape parameter vector \( p_k \) can be defined by Fourier series expansion [141] with \( N_f \) Fourier coefficients, i.e.,

\[
R(p, \phi) = R(\phi) \cdot p_k,
\]

where

\[
R(\phi) = \begin{bmatrix}
\frac{1}{2}, \cos(\phi), \sin(\phi), \ldots, \cos(N_F \phi), \sin(N_F \phi)
\end{bmatrix},
\]

\[
p_k = \begin{bmatrix}
\phi_0, \phi_1, \phi_2, \ldots, \phi_N
\end{bmatrix}^T.
\]

Fourier coefficients with small indices capture coarse shape features while coefficients with larger indices represent finer details.

The overall state vector \( x_k \) consists of the shape parameters \( p_k \), location \( d_k \), and kinematic parameters \( c_k \), i.e.,

\[
x_k = \begin{bmatrix}
p_k^T, d_k^T, c_k^T
\end{bmatrix}^T.
\]

A suitable measurement equation following the random hypersurface philosophy is formulated in polar form, i.e.,

\[
z_k = s_k \cdot r(p_k, \phi_k) + d_k + v_k
\]

where \( s_k \in [0, 1] \) is (multiplicative) noise that specifies the relative distance of the measurement source from the center, and \( \phi_k \) gives the angle to the measurement vector. In [6], it has been shown that \( s_k^2 \) is uniformly distributed in case the measurement sources are uniformly distributed over the shape. It can be approximated by a Gaussian distribution with mean 0.8 and covariance \( \frac{1}{12} \). By this means, the problem of estimating a (filled) shape has been reduced to a “curve fitting” problem, because for a fixed scaling factor \( s_k \), (20) specifies a closed curve. See also the discussion in Section 1.B.

The parameter \( \phi_k \) can be interpreted as a nuisance parameter (or latent variable) as in errors-in-variables models for regression and curve fitting. A huge variety of approaches for dealing with nuisance parameters has been developed in different areas. The most simple (and most inaccurate approach) is to replace the unknown \( \phi_k \) with a point estimate, e.g., the angle between \( d_k - z_k \) and the \( x \)-axis. This approach can be seen as greedy association model [33]. As the greedy association model yields to a bias in case of high noise, a so-called partial likelihood has been developed, which outperforms the greedy association model in many cases [33], [34], e.g., high noise scenarios. The partial likelihood model results from an algebraic reformulation of (20) and, hence, does not come with additional complexity, for further details see [33], [34].

A further natural approach would be to assume \( \phi_k \) to be uniformly distributed on the interval \([0, 2\pi]\), however, a nonlinear Kalman filter implicitly approximates a uniform distribution by a Gaussian distribution – a reasonable mean for this Gaussian approximation is not obvious due to the circular nature of \( \phi_k \).

Having derived the measurement equation (20), a measurement update can be performed using the formulas (18). As (20) is polynomial for given \( \phi_k \), closed-form formulas for the moments in (18) are available.

Due to the Gaussian state representation, closed-form formulas are available for linear dynamic models. For nonlinear dynamic models, nonlinear Kalman filters can be employed.

3) Random Hypersurface Model – Overview: In the same manner as for star-convex shapes [6], the concept of a random hypersurface model can be applied for circular and elliptic shapes [9]. In this case, it is more suitable to describe the shape with an implicit function instead of a parametric form. Unfortunately, representing the uncertainty of an elliptic shape with Gaussians is challenging – due to implicit constraints on the parameters.

Instead of using a Fourier series expansion for modeling the shape contour, [130] employ Gaussian processes within the RHM framework for star-convex shapes.

In many applications, the object to be tracked is symmetric, e.g., an aircraft or a vehicle. In this case specific improvements and adoptions can be performed in order incorporate symmetry information [32]. The concept of scaling the boundary of a curve in order to model an extended object has been combined with level sets in [140] in order to model arbitrary connected shapes. A closed-form likelihood for the use in nonlinear filters based on the RHM measurement equation (20) has been derived in [118].

The RHM idea can be used in the same manner to model three-dimensional shapes in three-dimensional space. In addition, two-dimensional shapes in three-dimensional space can also be modeled with RHM ideas [32], [35]. For example, in [32], measurements from a cylinder are modeled by means of translating a ground shape, i.e., a circle.

IV. TRACKING MULTIPLE OBJECTS

In this section we overview multiple extended object tracking. Regardless of object type—point, extended, group, etc—MTT is a problem that has many challenges:

- The number of objects is unknown and time varying.
- There are missed measurements, i.e., at each time step, all of the existing objects do not give measurements.
- There are clutter measurements, i.e., measurements that were not caused by a object.
- Measurement origin is unknown, i.e, the source of each measurement is unknown. This is often referred to as the “data association problem”.

For multiple point object tracking the literature is vast; a comprehensive overview of MTT algorithms, with a focus on point objects, was written by Vo et al. [129]. Many of the existing extended object MTT algorithms are of the RFS type. In the following subsections we will first introduce the
partitioning problem, which arises in extended object MTT as a part of the data association problem. We will then give a brief review of RFS filters, and lastly we will overview some examples of extended and group object MTT algorithms.

A. The partitioning problem

The possibility of multiple measurements per object means that the data association problem can be split into two parts:

1) Partitioning of the set of measurements into non-empty subsets called cells.

2) Association of the cells to the estimates.

An association from measurement to cell, and from cell to object estimate, defines an association from measurement to estimate.

In each partition, the cells are to be understood to contain measurements that are from the same source, i.e., either a object or clutter. For Bayes optimality it is necessary to consider all possible data associations, meaning that in extended and group MTT it is necessary to consider all possible partitions of the set of measurements. Indeed, summations over all possible partitions arise in many extended and group MTT filters, see, e.g., [15], [81], [88].

For a set of $N$ measurements, the number of possible partitions is given by the $N$th Bell number, see, e.g., [111]. For $N = 3$ measurements there are 5 possible partitions; an example is shown in Figure 5. The Bell numbers increase rapidly as $N$ increases; for twice the number of measurements ($N = 6$) there are 203 possible partitions, and for $N = 90$ measurements there are more than $10^{100}$ possible partitions. The result is an intractable computational complexity, and approximations are necessary for implementation.

An important contribution of [49], [51], [55] is to show how clustering can be used to find a subset of partitions. The basic insight behind the use of clustering lies in the definition of extended object: the measurements are spatially distributed around the object. Therefore spatially close measurements are more likely to be from the same object, than spatially distant measurements. By only considering a subset of partitions in which the cells contain spatially close measurements, the update becomes tractable.

Distance Partitioning [49], [51] is a clustering method that puts measurements in the same cell if the distance between a measurement and its closest neighbor is less than a threshold. By considering multiple thresholds, a subset of partitions is obtained. A detailed description of Distance Partitioning is given in [49], [51], and extensions of the method are proposed in [85]. An example is given in Figure 6. In this example there are 17 measurements, for which there are $> 10^{10}$ possible partitions. Using Distance Partitioning this is limited to 5 partitions.

Results from both simulations and experiments have shown that, despite the very drastic reduction in the number of partitions that are considered, performance is not sacrificed, see, e.g., [48], [53], [61], [114]. Finding a good subset of partitions is especially important when multiple extended objects are located in close vicinity of each other, see [15], [55], [81].

B. Review – RFS filters

A random finite set (RFS) is a set whose cardinality is a random variable, and whose set members are random variables. In RFS algorithms both the set of objects and the sets of measurements are modeled as RFSs. Tutorials on RFS methods can be found in, e.g., [63], [84], [127], and in-depth descriptions of the RFS concept and of finite set statistics (FISST) are given in the books [83], [85].

The set of objects that are present in the surveillance space is referred to as the multi-object state. Because of the computational complexity, specifically due to the data association problem, a full multi-object Bayes filter can be
quite computationally demanding to run, and typically approximations of the data association problem are necessary. Approximate filters include the Probability Hypothesis Density (PHD) filter [86], the Cardinalized PHD (CPHD) filter [87], and the various multi-Bernoulli (MB) filters, see, e.g., [83].

The first order moment of the multi-object state is called the PHD intensity, and can be said to be to a random set as the expected value is to a random variable. A PHD filter recursively estimates the PHD intensity under an assumed Poisson distribution for the cardinality.

A consequence of the Poisson assumption is that the PHD filter’s cardinality estimate has high variance, something that manifests itself, e.g., where there are missed measurements [30]. The CPHD filter recursively estimate the PHD and a truncated cardinality distribution, and is known to have a better insulation. The CPHD filter the objects are far enough away that they ought to be statistically independent.

In both the PHD filter and the CPHD filter the objects are independent identically distributed (iid); the normalized PHD intensity is the estimated object pdf. Both the PHD filter and the CPHD filter are susceptible to a “spooky effect” [42], [85], a phenomenon manifested by PHD intensity mass shifted from undetected objects to detected objects, even in cases when the objects are far enough away that they ought to be statistically insulated.

The MB-type filters approximate the multi-object density with a multi-Bernoulli distribution, which is then propagated and updated in time. In MB filters the objects are independent but not identically distributed, compared to the PHD and CPHD filters where the objects are iid. For each object, MB filters estimate a distribution for the object state, and a probability of existence. The MB filters are known for being capable of matching the CPHD filter’s cardinality performance without being susceptible to a “spooky effect”. There are also hybrid Poisson multi-Bernoulli filters, where the set of objects is divided into a set of detected objects, and a set of objects that have not yet been detected [136], [137].

Ultimately the desired output from an MTT algorithm is a set of estimated trajectories (tracks), where a trajectory is defined as the sequence of states from the time the object appears to the time it disappears. In their most basic forms neither the PHD nor the CPHD formally estimates object trajectories. However, trajectory estimates can be obtained, e.g., using post-processing with labelling schemes [52], [53], [99]. Alternatively labeled RFSs can be use, leading to the Generalized Labeled Multi-Bernoulli (GLMB) filter [128], and its computationally efficient approximation the Labeled Multi-Bernoulli (LMB) filter [106].

C. Examples of extended and group MTT

1) PHD filters: A PHD filter for extended objects under the Poisson model [45] was presented in [88]. Gaussian mixture implementations of this extended object PHD filter, for both linear and non-linear motion and measurement models, are presented in [49], [51]. The resulting filters can be abbreviated ET-GM-PHD filters. An approach to group object tracking based on a point object GM-PHD filter is presented in [21].

A Gaussian inverse Wishart implementation, using the random matrix extended object model [71] (Section III-A), is presented in [55], [57], and the resulting filter is abbreviated GIW-PHD filter. A Gaussian mixture implementation using RHMs (Section II-B) was presented in [142].

Multiple model Gaussian mixture PHD filters can be found in [48], [61]: the filters are applied to tracking of cars and bicycles, under assumed rectangle and stick shape models, and it is shown that using multiple measurement models can improve the estimation results.

Augmenting the implementations with gamma distributions makes it possible to estimate the unknown Poisson measurement rate for each object [55]. The resulting algorithms are then called gamma Gaussian (GG), or gamma Gaussian inverse Wishart (GGIW), respectively. The extended object PHD filter presented in [121], [122] is derived for a object model different from the Poisson point process model [45]. The objects are modelled by a Poisson cluster process, a hierarchic process with a parent process and a daughter process. The parent process models a Poisson distributed number of objects. For each object a daughter process models a number of reflection points that generate measurements. An implementation is proposed where the object is assumed ellipse shaped and the reflection points are located on the edge of the ellipse.

2) CPHD filters: A CPHD filter for extended objects was presented in [77], however the filter derivation is based on the quite strong assumption that “relative to sensor resolution, the extended objects and the unresolved objects are not too close and the clutter density is not too large” [77] Corollary 1. This assumption cannot be expected to hold in the general case. A CPHD filter capable of handling both spatially close objects and dense clutter is presented in [81], [95], [97], and a GGIW implementation is also presented. A comparison shows that the GGIW-CPHD filter outperforms the GGIW-PHD filter, especially when the probability of measurement is low, and/or the clutter density is high. The price for the increased performance is that the computational cost increases.

3) Multi-Bernoulli filters: Labeled MB filters for extended object tracking are presented in [14], [15], both a GLMB filter and its approximation the LMB filter. GGIW implementations are presented, and simulation results show that the labelled MB filters outperform their PHD and CPHD counterparts. Additionally, the GLMB and LMB filters estimate object trajectories, which the PHD and CPHD filters do only if labeling is used in post processing, see e.g., [52], [53]. A Poisson multi-Bernoulli filter for extended and group objects is presented in [47].

4) Non-RFS approaches: A Gaussian Mixture Markov Chain Monte Carlo filter for multiple extended object tracking is presented in [19]. The filter is compared to the linear ET-GM-PHD-filter [49], [51], and is shown to be less sensitive to clutter but also considerably more computationally costly (as measured by the average cycle time). The Probabilistic Multi-Hypothesis Tracker (PMHT) [119] allows more than one measurement per object, and the random matrix model (Section III-A) has been integrated in the PMHT framework, see [132], [134].

V. EXTENDED OBJECT TRACKING APPLICATIONS

Extended object tracking algorithms have been applied in many different scenarios and have been evaluated using
data from many different sensors such as LIDAR, camera, automotive radar, marine radar, GMTI radar, RGB-D sensors, and un-attended ground sensors (UGS). A list of references that contain experiments with real data is given in Table III.

In this section we will present three example applications:

- Tracking groups of pedestrians using a camera overlooking a footpath.
- Tracking cars using a LIDAR mounted in the grille of an autonomous vehicle.
- Tracking objects with complex shapes using an RGB-D sensor.

These three example are complementary in that they illustrate different aspects of extended object tracking: different sensor modalities; the applicability of extended object methods to group object tracking; object shapes of different complexities; and tracking in crowded scenarios with occlusions.

### TABLE III

**EXPERIMENTS WITH DIFFERENT SENSOR TYPES**

<table>
<thead>
<tr>
<th>Sensor</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIDAR</td>
<td>[15], [40], [41], [48], [50], [51], [55], [61], [79], [102]</td>
</tr>
<tr>
<td>Camera</td>
<td>[32], [52], [53], [115], [126], [133]</td>
</tr>
<tr>
<td>Marine Radar</td>
<td>[11], [29], [108], [109]</td>
</tr>
<tr>
<td>Automotive Radar</td>
<td>[46], [62]</td>
</tr>
<tr>
<td>RGB-D</td>
<td>[10], [12], [32], [35]</td>
</tr>
<tr>
<td>UGS, group tracking</td>
<td>[22]</td>
</tr>
<tr>
<td>Imaging Sonar</td>
<td>[22]</td>
</tr>
<tr>
<td>GMTI</td>
<td>[102]</td>
</tr>
</tbody>
</table>

### A. Tracking groups of pedestrians using camera

Automatic crowd surveillance is a complex task, and in scenes with a large number of persons it may be infeasible to track each person individually. In this case group object tracking is a viable alternative, as this does not require tracking and identification of each individual.

In the example presented here camera data is used to track groups of pedestrians that walk along a footpath. The online available PETS 2012 data set [124] is used for evaluation. For each image in the dataset a pedestrian detector [25], [26] is used, and the measurements are projected onto the ground plane using the camera parameters.

Groups of pedestrians are loosely constructed and typically do not have a detailed shape that remains constant over time. Therefore the groups can be assumed to be elliptically shaped, and the random matrix measurement model can be used [77]. The ground plane measurements are input into a GGIW-PHD filter [55], [56], and the object extractions are projected back into the camera image for visualization. The GGIW-PHD filter is based on the Poisson model for the number of measurements from each group, and for each group estimate a Poisson rate parameter is estimated. This estimated rate can be taken as an estimate of the number of persons in the group.

Example results are shown in Figure 74. The results show that the estimated ellipses are a good approximation of the pedestrian groups. The estimated Poisson rates tend to underestimate the number of persons in the group. The reason for this is that in groups with many persons, some individuals tend to be occluded and therefore are not detected. The estimated Poisson rate is more accurate when interpreted as a lower bound for the number of persons in the group, instead of interpreted as a count of the number of persons in the group.

### B. Tracking cars using LIDAR

Autonomous active safety features are standard in many modern cars, and in both research and industry there is a considerable push towards fully driverless vehicles. For safe operation in dense scenes, such as inner city and other urban environments, an autonomous vehicle must be capable of keeping track of other objects, to avoid collisions. To this end, high resolution sensors such as LIDAR and extended object tracking algorithms can be used.

The high angular resolution of LIDAR sensors typically results in a large number of measurements for each object. Thus, if an extended object tracking filter is not used, preprocessing is necessary to update the object estimates. Such preprocessing commonly consists of segmentation and clustering [67], [89], [104], shape fitting [91], or feature extraction [93]. The drawback of using such algorithms is that they are heavily dependent on parametrization, and often suffer from over- or under-segmentation. Especially in scenarios in which the environment changes, or when there are different object types, it is very difficult to find appropriate parameters. Because the tracking builds upon the data that is input, any error during segmentation and clustering will manifest itself as a tracking error.

In this section we will present experimental results where LIDAR sensors and an extended object PHD filter have been used to track cars; the results presented here are a subset of the results presented in [61]. The LIDAR sensor is assumed to be mounted in the grille of the ego vehicle, and the cars are assumed to be rectangular, with unknown length and width. The measurement modeling can be simplified by assuming that the LIDAR measurements are located along either one side of the assumed rectangular car, or along two sides. The measurement likelihood for these to cases are shown in Figure 8. The object dynamics are modeled using a so called bicycle model.

The tracking problem is cast as a multiple model problem, and a multiple model PHD filter is used to track multiple cars. A full description of the tracking algorithm can be found in [61]. When there are multiple cars in the sensor’s field of view the cars may occlude each other, either partially or fully. To avoid loosing track of cars that are occluded a non-homogeneous probability of measurement can be used. This is illustrated in Figure 9. Similar approaches to occlusion modeling is taken in [51], [55], [105], [138].

Experimental results in [61] show that the lateral position of the tracked cars can be estimated with an average error of less than 5cm, while the average longitudinal position error is slightly larger, around 10 to 30 cm for different datasets. The shape parameters are estimated with an average error

---

4 Video with tracking results: https://youtu.be/jN-KXQqargE
around 2 cm for the width, and around 20 cm for the length. Example tracking results for a scenario with four cars is given in Figure 10.

C. Tracking complex shapes using RGB-D sensor

In this subsection we present an experimental setup where complex object shapes are estimated using RGB-D sensor data. This experiment has been published first in [10]–[12]. Specifically, a moving miniature railway vehicle is to be tracked from a bird’s eye view with the help of a RGB-D camera. An optical flow algorithm determines the velocity of each image point in both the RGB and depth image sequences. Based on a threshold on the velocity, we obtain measurements, i.e., points classified as “moving”, that originate from the moving object. In this manner, a varying number of noisy mea-
Fig. 10. Results from scenario with four cars. Top: sensor data, color-coded according to time. Bottom: Estimated positions.

measurements/measurements from the object’s surface is received at each frame, see Figure [11] for an example frame. Due to the noisy images and inaccuracy of the optical flow algorithm, the measurements/measurements are noisy and do not completely fill the object surface. In fact, this is a typical extended object tracking problem where measurements come from a two-dimensional shape in two-dimensional space. Figure [12] shows example results with an implementation of the star-convex random hypersurface approach as discussed in Section III. Also, Figure [12] shows the result obtained from an active contour (snake) algorithm [69], which is a standard algorithm in computer vision. In general, an active contour model works with intensity/RGB images and not with point measurements. It calculates a contour by minimizing an energy function [69] that is composed of an external force for pushing the contour to image features and an internal force for regularization. In this scenario, active contours are applied to the depth image and hence, can be unreliable in case the vehicle passes objects with similar depth, see Figure [12].

Alternatively, active contours can be applied to a “smoothed” version of the point measurements: the measurements are interpreted as an intensity image by placing a Gaussian kernel function at each measurement location. As indicated by Figure [13], active contours then aim at determining an enclosing curve of the point measurements in each frame. As the vehicle’s surface is not covered completely by the measurements in a single frame, active contours do not give a reasonable shape estimate. Active contours are not capable of systematically accumulating individual point measurements over time – without this capability no reasonable shape estimate can be expected.

VI. SUMMARY AND CONCLUDING REMARKS

In this article we gave an introduction to extended object tracking, a comprehensive up-to-date overview of state-of-the-art research, and illustrated using several different sensors and object types. Increasing sensor resolutions mean that there will be an increasing number of scenarios in which extended object methods can be applied. It is possible to cluster/segment the data in pre-processing and then apply standard point object methods, however this requires careful parameter tuning,
thereby increasing the risk for errors. Extended object tracking, on the other hand, uses Bayesian models for the multiple measurements per object, meaning that the tracking performance is much less dependent on clustering/segmentation.

During the last 10 years an impressive number of new methods and applications have appeared in the literature, covering different approaches to extent modeling and multiple object tracking. This trend that can be expected to continue, as there are many open questions to solve, and improvements can be made. Due to the high non-linearity and high dimensionality of the problem, estimation of arbitrary shapes is still very much challenging. There is a need for performance bounds for extended object tracking methods: for a given shape model, how many measurements are required in order for the estimation algorithm to converge to an estimate with small error? Such performance bounds may help in answering the question of which shape complexity is suitable when modeling the object. Naturally, in most applications one is interested in a shape description that is as precise as possible.

REFERENCES


