Exploiting Negative Measurements for Tracking Star-Convex Extended Objects

Antonio Zea, Florian Faion, Jannik Steinbring, and Uwe D. Hanebeck[1]

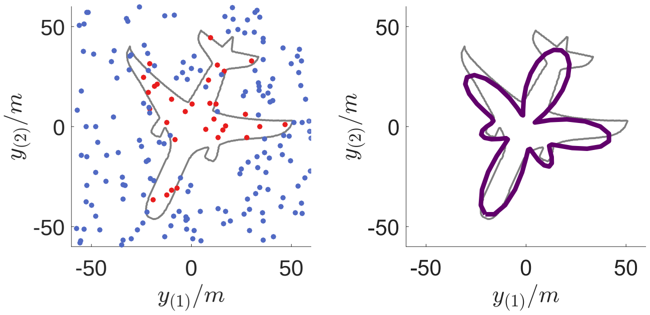
***Abstract*— In this paper, we propose a novel approach to track extended objects by incorporating negative information. While traditional techniques to track extended targets use only *positive* measurements, assumed to stem from the target, the proposed estimator is also capable of incorporating *negative* measurements, which tell us where the target cannot be. To achieve this, we introduce a simple, robust, and easy-to-implement recursive Bayesian estimator which employs ideas from the field of curve fitting. As an application of this idea, we develop a measurement equation to estimate star-convex shapes which can be used in standard non-linear Kalman filters. Finally, we evaluate the proposed estimator using synthetic data and demonstrate its robustness in scenarios with clutter and low measurement quality.**

I. Introduction

While classical tracking approaches consider the target to be a single point, increasing sensor resolution allows for multiple source points on the target object to be measured. This allows for the development of more accurate and robust estimators by taking into account the shape of the target. However, the extent may be unknown, leading to the problem of estimating the shape and pose parameters of the target simultaneously. Treating this problem defines the field of extended object tracking (EOT).

Literature provides a multitude of approaches that deal with EOT. In case of low available information, the target is approximated as a simple shape such as lines [1], [2] or polynomials [3]. A popular branch consists of approximations using ellipses based on random matrices [4]–[6]. As more information becomes available, the shape can become more detailed. For instance, some approaches parameterize the boundary radially from a central point, using Fourier series [7], extended Gaussian images [8], or Gaussian processes [9]. As it deals with shapes, EOT can incorporate many ideas of distance minimization and curve fitting [10], [11], as explored in [7], [12], [13].

On a related line, tracking techniques can also be categorized based on the kinds of measurements they can process. The general approach is to exploit measurements that stem from the target, denoted as *positive* measurements. However, *negative* measurements, which are known not to stem from the object, are also valuable as they indicate where the target cannot possibly be, even if they are generally discarded as useless clutter. Exploiting negative measurements for tracking has also been treated in [14]–[16]. In cases of low measurement quality, this additional information can become extremely valuable, and may even keep the estimator from diverging.



(a) Positive, negative measurements.

(b) Example star-convex estimate.

Fig. 1: Positive measurements (red) show where the target is, while negative measurements (blue) show where the target *cannot* be. We propose a robust and easy-to-implement Bayesian estimator for star-convex shapes (purple) that can incorporate both kinds of information.

The authors of this work have previously proposed a model to incorporate negative observations for extended object tracking in [17], which allowed for the derivation of an explicit likelihood which could be used, e.g., in particle filters [18]. In this paper, we will propose an alternative formulation that employs ideas from curve fitting (Fig. 1). The main contribution is the derivation of a measurement equation for star-convex shapes, which can be used in non-linear Kalman filters such as the UKF [19]. This approach results in a robust estimator that is easy to implement. However, an explicit likelihood based on this new model will also be provided for applications that need it.

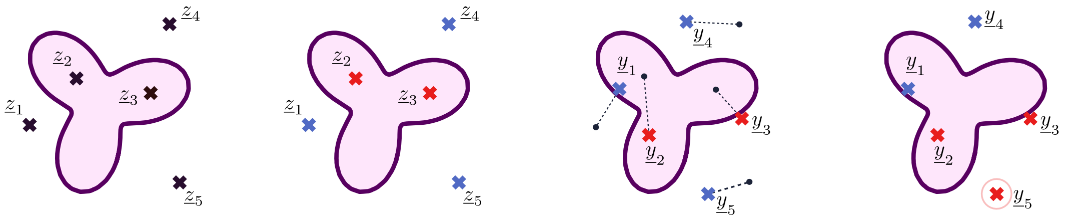
This paper is structured as follows. Sec. II presents a formal formulation of the discussed problem. Then, Sec. III describes how extended objects are traditionally modeled. The proposed approach is introduced in Sec. IV, and Sec. V presents an evaluation of the implementation. Finally, Sec. VI concludes the paper.

II. Problem Formulation

The task being considered is estimating the state, in particular the shape and pose, of a target object. At the timestep k, the state parameters are described as the vector $\underline{x}_k$. The shape of the target is denoted as the set of points $\mathcal{S}_k \subset \mathbb{R}^n$. For this paper, we will focus on shapes in $\mathbb{R}^2$. Furthermore, we assume that the shape is *filled*, i.e., the target consists of the shape boundary and its interior.

In order to estimate the state, we incorporate sensor measurements which are assumed to take the following form.

[1] The authors are with the Chair of Intelligent Sensor-Actuator-Systems (ISAS) at the Karlsruhe Institute of Technology (KIT), Germany. antonio.zea@kit.edu, florian.faion@kit.edu, jannik.steinbring@kit.edu, uwe.hanebeck@ieee.org



(a) Selection of sources. (b) Determining type. (c) Noise on position. (d) Noise on type.

Fig. 2: Generative model for measurements. First, a set of source positions is drawn from $\mathbb{R}^2$, and their corresponding types are set as positive if they are inside the shape (red), or negative if outside (blue). Then, during observation, the measurements are subject to corruption by noise, both in their positions (dotted lines) and in their types (red circle).

At the timestep k, we obtain a set of position measurements $\{\underline{y}_{k,1}, \cdots, \underline{y}_{k,n_k}\}$ in Cartesian coordinates, and a corresponding set of measurement types $\{\tau_{k,1}, \cdots, \tau_{k,n_k}\}$. It is not assumed that all measured positions belong to the target shape. Instead, for a position $\underline{y}_{k,i} \in \mathbb{R}^2$, the corresponding type $\tau_{k,i}$ can take one of two values. If $\tau_{k,i} = \odot^+$, it is assumed that the positive measurement $\underline{y}_{k,i}$ originates from the target. However, if $\tau_{k,i} = \odot^-$, we consider that the negative measurement $\underline{y}_{k,i}$ does *not* stem from the target. Measurement types can be obtained, for example, by segmentation using state-independent information. Fig. 3 shows a practical example taken from a real sensor.

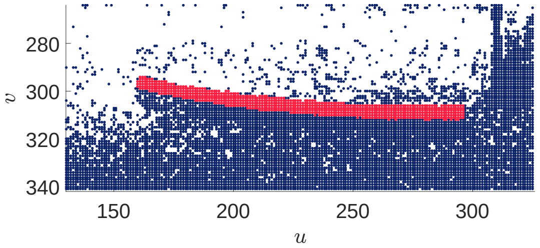


Fig. 3: A real-life target object captured by a Kinect sensor. Using RGB and planar segmentation, measurements were classified as either positive (red) or negative (blue). White gaps represent missing information due to sensor artifacts.

More formally, the observed measurements are assumed to have been the result of the following generative model. First, a set of source positions $\{\underline{z}_{k,1}, \cdots, \underline{z}_{k,n_k}\}$ is drawn from $\mathbb{R}^2$ (Fig. 2a) using an arbitrary but unknown rule. Then, based on the state $\underline{x}_k$, a type $t_{k,i}$ is associated to each position $\underline{z}_{k,i}$ in the following way. If $\underline{z}_{k,i}$ is inside of the target shape, the corresponding type $t_{k,i}$ is set as $\odot^+$. Otherwise, it is set to $\odot^-$ (Fig. 2b). Finally, during observation by the sensor, errors are introduced into each measurement which affect their position and type. The position is corrupted by additive zero-mean Gaussian noise, yielding

$$\underline{y}_{k,i} = \underline{z}_{k,i} + \underline{v}_{k,i} \ ,$$

where $\underline{v}_{k,i} \sim \mathcal{N}(\underline{0}, \mathbf{R}_{k,i})$ (Fig. 2c). The measurement type $t_{k,i}$ is also corrupted. A positive type has a probability p_{FN} of being detected as a false negative, and a negative type has a probability p_{FP} of turning into a false positive (Fig. 2d), yielding the observed type $\tau_{k,i}$

In probabilistic terms, this generative model can be described using the conditional probability density $\mathrm{p}(\underline{y}_{k,i}, \tau_{k,i} \,|\, \underline{x}_k)$. We assume that the noise terms are independent from each other, so that the probability density for all combined measurements becomes

$$\mathrm{p}(\underline{y}_{k,1}, \tau_{k,1}, \cdots, \underline{y}_{k,n_k}, \tau_{k,n_k} \,|\, \underline{x}_k) = \prod_{i=1}^{n_k} \mathrm{p}(\underline{y}_{k,i}, \tau_{k,i} \,|\, \underline{x}_k) \ .$$

This allows us to treat each measurement individually in the following sections, as multiple measurements can be combined by multiplying their probabilities sequentially. For this reason, we will drop the subindex i unless needed.

Finally, the state may evolve in time from $\underline{x}_k$ to $\underline{x}_{k+1}$ using a system model. Note that this paper is not concerned with the state evolution, and imposes no constraints on the system model.

III. Modeling Extended Objects

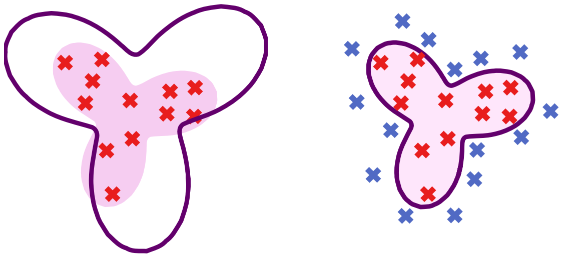
In this section, we will give a short description of how extended targets are usually modeled, and a motivation for the new proposed approach. Note that traditional models generally take only into account positive measurements, and thus, it is assumed that all measurements stem from the shape and $\tau_k = \odot^+$. The defining aspect is how to derive the probability density $\mathrm{p}(\underline{y}_k \,|\, \underline{x}_k)$ which relates the measurement $\underline{y}_k$ to the state $\underline{x}_k$. The importance of this term stems from the fact that, by interpreting it as a likelihood function, it allows for the derivation of a Bayesian estimator. However, this task is made difficult by the fact that we do not know which source $\underline{z}_k$ generated $\underline{y}_k$ due to the measurement noise. This crucial issue is known as the *association problem.*

A commonly used approach to address this challenge is by using *Spatial Distribution Models* (SDMs) [1], [20]. The idea is to associate $\underline{y}_k$ to all possible sources using a probability density $\mathrm{p}(\underline{z}_k \,|\, \underline{x}_k)$, which assigns to each point in $\mathbb{R}^2$ a probability to generate a measurement. However, the resulting term $\mathrm{p}(\underline{y}_k \,|\, \underline{x}_k)$ takes the form of a difficult integral for which closed-form solutions are generally unavailable, raising the need for different simplifications, such as [4].

Furthermore, in cases of occlusions or sensor artifacts, it may become impossible to obtain an appropriate approximation for the source probability, which leads to estimation bias.

A different idea is to associate $\underline{y}_k$ to a single source $\underline{\tilde{z}}$, calculated as the point in $\mathcal{S}_k$ which minimizes some sort of distance function to $\underline{y}_k$. Then, the distance to this assumed source is used as a pseudo-measurement, and hence, the estimation task is reduced to a distance-minimization problem. We denote these approaches as *Greedy Association Models* (GAMs) [13], which can be seen as a probabilistic interpretation of traditional curve fitting techniques [10], [11]. Due to considering only a single source, the term $\mathrm{p}(\underline{y}_k \,|\, \underline{x}_k)$ required for this model is simple, fast, and easy to evaluate. Furthermore, as no explicit source probability is required, these approaches are generally much more robust against occlusion [13].

However, GAMs tend to suffer from an important issue, which appears especially in shapes that also generate measurements from their interior, such as disks [12] or line segments [2]. The problem is that, if there are multiple states whose shapes contain the target object, the estimator cannot discern between them, as all of them also contain the received measurements (Fig. 4a), and thus, the distances to these shapes are also 0. In some cases, this will cause the estimator to diverge, as the estimated shape can become arbitrarily large. We denote this issue as the *length problem*. As shown in Fig. 4b, incorporating negative measurements would address this issue, by imposing a constraint on both sides of the boundary. This is the key idea behind the approach introduced in the following section.



(a) Length problem.

(b) Proposed approach.

Fig. 4: The length problem appears, e.g., in filled shapes, when multiple shapes contain all possible measurements (left), and thus, the optimal state is not unique. By also incorporating negative measurements (right), the optimal state is unambiguous.

IV. Exploiting Negative Measurements

In this section, we will introduce a recursive Bayesian estimator to track extended objects by incorporating both positive and negative measurements. While an approach based on SDMs was proposed in [17], in this paper we will take a different direction and exploit ideas of GAMs and curve fitting, avoiding the pitfalls described in Sec. III. Furthermore, in order to show the applicability of the concepts, and for illustration, we will also describe an example implementation that approximates targets as filled star-convex shapes. We will also show how these techniques can also be applied for arbitrary shapes.

This section is structured as follows. First, we formally derive a recursive Bayesian estimator. Then, we proceed to define the characteristics of the star-convex shapes, and the functions we will need for the update step. Finally, we describe how the update step can be implemented.

A. Deriving a Bayesian Estimator

A Bayesian estimator maintains knowledge about the state $\underline{x}_k$ in the form of a probability density. It consists of two steps. On the one hand, the *update* step corrects the knowledge about the state based on incoming measurements. On the other hand, the *prediction* step lets the state evolve in time using a system model. As mentioned in Sec. II, this paper is only concerned with the update step, and imposes no restrictions or constraints on the prediction step.

The update step of the Bayesian estimator can be derived by interpreting the conditional density $\mathrm{p}(\underline{y}_k, \tau_k \,|\, \underline{x}_k)$ as a likelihood function. Using Bayes' rule, we obtain

$$\begin{aligned} f_k^e(\underline{x}_k) &= \mathrm{p}(\underline{x}_k \,|\, \underline{y}_k, \tau_k) \\ &= \frac{\mathrm{p}(\underline{y}_k, \tau_k \,|\, \underline{x}_k)}{\mathrm{p}(\underline{y}_k, \tau_k)} \cdot f_k^p(\underline{x}_k) \\ &= \underbrace{\frac{\mathrm{p}(\tau_k \,|\, \underline{x}_k)}{\mathrm{p}(\tau_k)}}_{\text{Type update}} \cdot \underbrace{\frac{\mathrm{p}(\underline{y}_k \,|\, \underline{x}_k, \tau_k)}{\mathrm{p}(\underline{y}_k \,|\, \tau_k)}}_{\text{Position update}} \cdot f_k^p(\underline{x}_k) \ . \end{aligned}$$

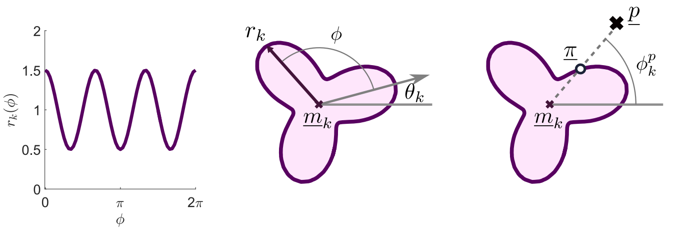
where $f_k^p(\underline{x}_k)$ is denoted as the prior, and $f_k^e(\underline{x}_k)$ as the posterior distribution. We observe that this step can be divided into two partial updates, the type update which only considers the measurement type, and the position update which only considers the position information.

In literature, there are two main filtering techniques to implement the update step. The first is by means of particle filters, which use the likelihood explicitly. The second one is using *Linear Regression Kalman Filters* (LRKFs), such as [19], [21], which represent the prior as $\mathcal{N}(\underline{x}_k; \hat{\underline{x}}_k^p, \mathbf{C}_k^p)$, and approximate the posterior as $\mathcal{N}(\underline{x}_k; \hat{\underline{x}}_k^e, \mathbf{C}_k^e)$. These do not work directly with the likelihood, but employ a measurement equation instead. The following implementation will show how to apply the proposed concepts for both types of estimators, so that an application can choose whichever fits it best.

As an aside, we note that the number of received measurements also contains information about the state. The likelihood function can be extended to incorporate this information as explored in [4], [20], [22]. For reasons of brevity and space, we decided to leave this discussion out of this paper.

B. Star-Convex Shapes using Fourier Series

We will now describe the characteristics of the star-convex shapes. The boundary of a star-convex shape of degree n_F is defined by a center $\underline{m}_k$, a rotation θ_k, and a sequence of coefficients $\{a_k^{(0)}, a_k^{(1)}, b_k^{(1)}, \cdots, a_k^{(n_F)}, b_k^{(n_F)}\}$. We assume that these parameters are contained in the state vector $\underline{x}_k$. We



(a) Distance $r_k(\phi)$. (b) Star-convex shape. (c) "Closest" point $\underline{\pi}$.

Fig. 5: Example star-convex shape with center $\underline{m}_k$, rotated by θ_k. Similar to polar coordinates, $r_k(\phi)$ determines the radial distance to the boundary (dark purple) at the angle ϕ.

define the radial distance function (Fig. 5a) as the Fourier series

$$r_k(\phi) := \frac{a_k^{(0)}}{2} + \sum_{j=1}^{n_F} a_k^{(j)} \cos(j\phi) + b_k^{(j)} \sin(j\phi) \enspace .$$

Then, the shape $\mathcal{S}_k$ (Fig. 5b) can be constructed as the set

$$\mathcal{S}_k := \{\underline{m}_k + s \cdot r_k(\phi)\, \underline{e}(\phi + \theta_k) \mid s \in [0,1], \phi \in [0, 2\pi]\}$$

where $\underline{e}(\alpha) := \left[\cos(\alpha), \sin(\alpha)\right]^T$. Furthermore, we define the *boundary* of $\mathcal{S}_k$ as all points generated for $s = 1$. It follows that this shape has the geometric property of being *star-convex*, that is, we can connect all points in the shape to the center point using a segment that is completely inside the shape.

We define the following convenience functions. Given a point $\underline{p} = \left[p_0, p_1\right]^T \in \mathbb{R}^2$, the angle between the segment that connects it the origin and the x-axis is

$$\angle\left(\underline{p}\right) := \operatorname{atan2}(p_1, p_0) \enspace ,$$

where $\operatorname{atan2}(\cdot,\cdot)$ is the four-quadrant inverse tangent. Then, we define $\phi_k^p := \angle\left(\underline{p} - \underline{m}_k\right)$ as the direction of $\underline{p}$. Finally, we define the function

$$\underline{\pi}(\underline{x}_k, \underline{p}) := \underline{m}_k + r_k\left(\phi_k^p - \theta_k\right) \cdot \underline{e}\left(\phi_k^p\right) \enspace ,$$

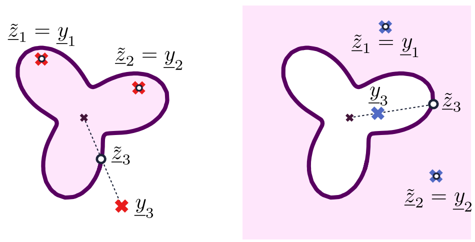
which yields the point on the shape boundary in the direction ϕ_k^p. We denote this as the "closest" boundary point to $\underline{p}$ (Fig. 5c). Note that this point is not necessarily the closest in the sense of the Euclidean distance, and is chosen only for its ease of calculation.

C. Shape-related Functions

For the following subsections, we will require three functions related to the shape. Note that these functions are not specific to star-convex shapes, and can be re-implemented to describe other shapes. The subsequent update step can still be applied in the same way.

We consider an arbitrary measurement $\underline{y}_k \in \mathbb{R}^2$ of type τ_k. First, we need a function $\operatorname{Inside}(\underline{x}_k, \underline{y}_k)$ that tells us whether $\underline{y}_k$ is inside $\mathcal{S}_k$. For star-convex shapes, this follows as

$$\operatorname{Inside}(\underline{x}_k, \underline{y}_k) \leftrightarrow \left[\|\underline{y}_k - \underline{m}_k\| \leq r_k\left(\phi_k^y - \theta_k\right)\right] \enspace ,$$



$\underline{\tilde{z}}_1 = \underline{y}_1$
$\underline{\tilde{z}}_2 = \underline{y}_2$
$\underline{\tilde{z}}_3$
$\underline{y}_3$
$\underline{\tilde{z}}_1 = \underline{y}_1$
$\underline{y}_3$
$\underline{\tilde{z}}_3$
$\underline{\tilde{z}}_2 = \underline{y}_2$

(a) Positive measurements. (b) Negative measurements.

Fig. 6: Assumed source. For positive measurements, the source is the closest point in $\mathcal{S}_k$. For negative measurements, the source is the closest point outside $\mathcal{S}_k$, or on its boundary.

where $\|\cdot\|$ is the Euclidian norm. This means that $\underline{y}_k$ is inside iff it is closer to the center $\underline{m}_k$ than the boundary in that direction. Second, we need to calculate an assumed source $\underline{\tilde{z}}(\underline{x}_k, \underline{y}_k, \tau_k)$. We define it as

$$\underline{\tilde{z}}(\underline{x}_k, \underline{y}_k, \tau_k) = \begin{cases} \underline{y}_k & \text{if } \tau_k = \odot^+ \wedge \operatorname{Inside}(\underline{x}_k, \underline{y}_k) \\ & \text{ or } \tau_k = \odot^- \wedge \neg \operatorname{Inside}(\underline{x}_k, \underline{y}_k) \\ \underline{\pi}(\underline{x}_k, \underline{y}_k) & \text{otherwise.} \end{cases}$$

The idea behind this function is simple. If $\underline{y}_k$ is of type $\odot^+$, we assume it was generated on the inside of $\mathcal{S}_k$. Thus, if it is already inside, the closest point is $\underline{y}_k$ itself. Otherwise, we return the closest point on the boundary (Fig. 6a). Similarly, if $\underline{y}_k$ is of type $\odot^-$ and it is already outside, we return $\underline{y}_k$ itself, otherwise, we calculate the closest point on the boundary (Fig. 6b).

Third, and finally, we need a function $\Phi(\underline{x}_k, \underline{y}_k, \tau_k)$ that calculates the distance between $\underline{y}_k$ and its assumed source. This follows simply as

$$\Phi(\underline{x}_k, \underline{y}_k, \tau_k) := \|\underline{y}_k - \underline{\tilde{z}}(\underline{x}_k, \underline{y}_k, \tau_k)\| \enspace . \tag{1}$$

D. The Position Update

As is common in GAMs, we will use the distance to the shape as a pseudo-measurement. We observe that, if we remove the effect of noise, the measured positions always match their assumed sources, and the distances generated by (1) always equal 0. This leads to the measurement equation

$$0 \;= \Phi(\underline{x}_k, \underline{y}_k - \underline{v}_k, \tau_k) \enspace . \tag{2}$$

However, [13] showed that this sort of measurement equation produces a biased result, and proposed a mechanism to alleviate this bias, which works as follows. We rewrite (2) as

$$0 \;= \Phi(\underline{x}_k, \underline{y}_k, \tau_k) - \epsilon_k \enspace , \tag{3}$$

where ϵ_k is a new noise term derived from $\underline{v}_k$, defined as

$$\epsilon_k := \Phi(\underline{x}_k, \underline{z}_k + \underline{v}_k, \tau_k) \enspace . \tag{4}$$

For convenience and speed, we will approximate ϵ_k as follows. First, we will only consider the case for $\underline{x}_k \approx \hat{\underline{x}}_k^p$, i.e., the prior mean. Second, we approximate $\underline{z}_k \approx \underline{\tilde{z}}(\hat{\underline{x}}_k^p, \underline{y}_k, \tau_k)$. Then, we calculate the mean $\hat{\epsilon}_k$ and variance $\sigma_{\epsilon_k}^2$ of the new noise term by propagating $\underline{v}_k$ through (4), as described, for example, in Algorithm 1. Note that in all algorithms we drop the subindex k for legibility. Finally, by moment matching, we assume that $\epsilon_k \sim \mathcal{N}(\hat{\epsilon}_k, \sigma_{\epsilon_k}^2)$. The likelihood for particle filters follows from (3) as

$$\mathrm{p}(\underline{y}_k \,|\, \underline{x}_k, \tau_k) \approx \mathcal{N}(\Phi(\underline{x}_k, \underline{y}_k, \tau_k); \hat{\epsilon}_k, \sigma_{\epsilon_k}^2) \ .$$

Algorithm 1: Calculate additive noise moments.

```
  input : $\underline{\hat{x}}^p, \underline{y}, \tau$,
          noise samples with weights $\{\underline{\nu}^j, W^j\}_{j=1}^{n_v}$
1 $\underline{z} \leftarrow \underline{\tilde{z}}(\underline{\hat{x}}^p, \underline{y}, \tau)$ ;
2 for $j = 1$ to $n_v$ do
3     $\epsilon^j \leftarrow \Phi(\underline{\hat{x}}^p, \underline{z} + \underline{\nu}^j, \tau)$ ;
4 $\hat{\epsilon} \leftarrow \sum_{j=1}^{n_v} W^j \cdot \epsilon^j$ ;
5 $\sigma_\epsilon^2 \leftarrow \sum_{j=1}^{n_v} W^j \cdot \left[\epsilon^j - \hat{\epsilon}\right]^2$ ;
  output : $\hat{\epsilon}, \sigma_\epsilon^2$
```

For LRKFs, the update step can be implemented using Algorithm 2, using the state samples (also called *sigma points*) provided by the filter, and the well-known Kalman formulas.

Algorithm 2: Position update step for LRKFs.

```
  input : $\underline{\hat{x}}^p, \mathbf{C}^p, \underline{y}, \tau$,
          prior state samples with weights $\{\underline{\chi}^j, W^j\}_{j=1}^{n_x}$
1 $\hat{\epsilon}$, $\sigma_\epsilon^2 \leftarrow$ Calculate moments using Algorithm 1 ;
2 for $j = 1$ to $n_x$ do
3     $\varphi^j \leftarrow \Phi(\underline{\chi}^j, \underline{y}, \tau)$ ;
4 $\hat{\varphi} \leftarrow \sum_{j=1}^{n_x} W^j \cdot \varphi^j$ ;
5 $\sigma_\varphi^2 \leftarrow \sum_{j=1}^{n_x} W^j \cdot \left[\varphi^j - \hat{\varphi}\right]^2$ ;
6 $\mathbf{C}^{x\varphi} \leftarrow \sum_{j=1}^{n_x} W^j \left[\underline{\chi}^j - \underline{\hat{x}}^p\right] \left[\varphi^j - \hat{\varphi}\right]$ ;
7 $\mathbf{K} \leftarrow \mathbf{C}^{x\varphi} \cdot (\sigma_\epsilon^2 + \sigma_\varphi^2)^{-1}$ ;
8 $\underline{\hat{x}}^e \leftarrow \underline{\hat{x}}^p + \mathbf{K} \cdot (\hat{\epsilon} - \hat{\varphi})$ ;
9 $\mathbf{C}^e \leftarrow \mathbf{C}^p - \mathbf{K} \cdot (\sigma_\epsilon^2 + \sigma_\varphi^2) \cdot \mathbf{K}^T$ ;
  output : $\underline{\hat{x}}^e, \mathbf{C}^e$
```

E. The Type Update

The second part of the update step only takes into account the measurement type. However, calculating $\mathrm{p}(\tau_k \,|\, \underline{x}_k)$ directly is difficult, given that it depends on the probability for each point in space to generate a measurement, which is generally unknown. Instead, using the same ideas as Sec. IV-D, we will focus only on the neighborhood of the assumed measurement source, and calculate how likely it would be to generate a measurement of the given type. Following this idea, the term $\mathrm{p}(\tau_k \,|\, \underline{x}_k)$ can be calculated using Algorithm 3.

This result can be used directly as an explicit likelihood for particle filters. For LRKFs, a measurement equation is not directly available, and the usual update mechanism is inadequate given that τ_k is a discrete term. Instead, an alternative update mechanism inspired by [23] is provided in Algorithm 4. Note that, as usual in particle-based approaches, this mechanism may have issues with sample degeneration. Thus, if the number of non-zero likelihoods is smaller than the state dimension, the measurement should be discarded.

Algorithm 3: Type likelihood for a given state $\underline{x}$.

```
   input : $\underline{x}, \underline{y}, \tau, \mathrm{p}_{FN}, \mathrm{p}_{FP}$,
           noise samples with weights $\{\underline{\nu}^j, W^j\}_{j=1}^{n_v}$
1  $\underline{z} \leftarrow \underline{\tilde{z}}(\underline{x}, \underline{y}, \tau)$ ;
2  $\mathrm{p}^+ \leftarrow 0$ ;
3  for $j = 1$ to $n_v$ do
4      if Inside($\underline{x}, \underline{z} + \underline{\nu}^j$) then
5          $\mathrm{p}^+ \leftarrow \mathrm{p}^+ + W^j$ ;
6  $\mathrm{p}^- \leftarrow 1 - \mathrm{p}^+$ ;
7  if $\tau = \odot^+$ then
8      $\ell \leftarrow (1 - \mathrm{p}_{FN}) \cdot \mathrm{p}^+ + \mathrm{p}_{FP} \cdot \mathrm{p}^-$ ;
9  else
10     $\ell \leftarrow \mathrm{p}_{FN} \cdot \mathrm{p}^+ + (1 - \mathrm{p}_{FP}) \cdot \mathrm{p}^-$ ;
   output : $\ell$
```

Algorithm 4: Type update step for LRKFs.

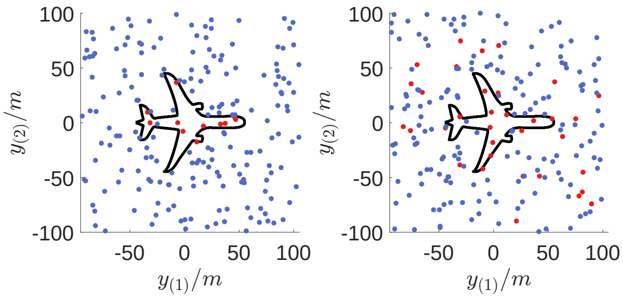
```
  input : $\underline{y}, \tau$,
          prior state samples with weights $\{\underline{\chi}^j, W^j\}_{j=1}^{n_x}$
1 for $j = 1$ to $n_x$ do
2     $\ell^j \leftarrow$ Likelihood for $\underline{\chi}^j$ using Algorithm 3 ;
3 for $j = 1$ to $n_x$ do
4     $\overline{W}^j \leftarrow W^j \cdot \ell^j / \left[\sum_{i=1}^{n_x} W^i \cdot \ell^i\right]$ ;
5 $\underline{\hat{x}}^e \leftarrow \sum_{j=1}^{n_x} \overline{W}^j \cdot \underline{\chi}^j$ ;
6 $\mathbf{C}^e \leftarrow \sum_{j=1}^{n_x} \overline{W}^j \cdot \left[\underline{\chi}^j - \underline{\hat{x}}^e\right] \left[\underline{\chi}^j - \underline{\hat{x}}^e\right]^T$ ;
  output : $\underline{\hat{x}}^e, \mathbf{C}^e$
```

F. Gating

In real scenarios, it is often the case that $\mathrm{p}_{FN} \neq 0$ or $\mathrm{p}_{FP} \neq 0$, and thus, it becomes necessary to find a mechanism to reject clutter measurements that are far from the boundary, while also taking into account the measurement noise and the state uncertainty. For LRKFs, we propose the following straightforward mechanism using ideas from validation gates for Kalman filters [24] that can be incorporated into Algorithm 2. Thus, we discard the measurement if

$$(\hat{\epsilon} - \hat{\varphi})^2 \cdot (\sigma_\epsilon^2 + \sigma_\varphi^2)^{-1} > G \ .$$

This step can be introduced after Line 5. The left term can be approximated as having a χ^2 distribution with one degree of freedom, which allows an intuitive selection of the gating parameter G. For example, if we want to accept 99% of



(a) No clutter.

(b) $\mathrm{p}_{FN} = \mathrm{p}_{FP} = 0.1$.

Fig. 7: Example snapshots with 200 total measurements, without and with clutter. Measurement noise is $\mathbf{R}_k = 4 \cdot \mathbf{I}\,\mathrm{m}^2$.

valid measurements, we use $G = \mathrm{chi2inv}(0.99, 1) \approx 6.6349$, where $\mathrm{chi2inv}(\cdot, 1)$ is the inverse function of the $\chi^2(1)$ CDF.

V. Evaluation

In this section, we will present an evaluation of the proposed concepts using synthetic data. The setup is as follows. The state has the form

$$\underline{x}_k = \left[\underline{m}_k^T, \theta_k, \mathrm{v}_k, \omega_k, a_k^{(0)}, \cdots, a_k^{(n_F)}, b_k^{(n_F)}\right]^T ,$$

where $n_F = 7$, leading to a state of dimension 20. As in Sec. IV-B, $\underline{m}_k \in \mathbb{R}^2$ is the shape center, θ_k the rotation, and the values $a_k^{(\cdot)}$ and $b_k^{(\cdot)}$ represent the Fourier coefficients. The target is assumed to be moving in the direction of θ_k at a speed of v_k, and turning with rotational velocity ω_k.

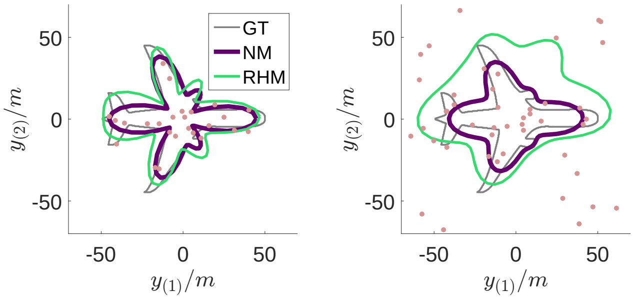
At each timestep k, 200 measurements are generated uniformly from the sensor's field of view, according to the generative model from Sec. II. An example setup is shown in Fig. 7. The number of resulting measurements that originate from the target varies every timestep, approximately following a Poisson distribution with mean 11. Note that the total number of positive measurements also depends on the number of false positives determined by p_{FP}.

The estimator to be used is the S^{2}KF [21], following the LRKF implementation proposed in Algorithm 2 and Algorithm 4. The number of state samples is $16 \times 20 = 320$. The number of measurement noise samples for Algorithm 1 is 21, also taken from the S^{2}KF. The update step is extended to implement the validation gate of 99% proposed in Sec. IV-F. The process noise is assumed to be white, additive, and distributed as $\underline{w}_k \sim \mathcal{N}(\underline{0}, \mathbf{Q})$, where $\mathbf{Q} = 10^{-1} \cdot \mathrm{diag}(1, 1, 10^{-4}, 1, 10^{-4}, 1, 1, \cdots, 1)$. Measurements are processed sequentially as they arrive, and the process noise is applied only at the end of each timestep.

The proposed approach will be compared against the star-convex *Random Hypersurface Model* (RHM) described in [7], using the S^{2}KF. The RHM scaling factor is approximated as $\mathcal{N}(\frac{2}{3}, \frac{1}{18})$. All other parameters, including the gating mechanism, are identical for both approaches.

A. Static Target

For the static target evaluation, the target is assumed not to be moving. The initial position is $\left[15, 15\right]^T$. For the Fourier coefficients, the values were set as $a_0^{(0)} = 60$, and the rest as 0. This represents a circle of radius 30 m. Two experiments were started, without and with clutter. The results are shown in Fig. 8. Ground truth is gray, the proposed model ("NM") in purple, and the RHM in green. The illustrated shapes correspond to the final states averaged over 50 runs.



(a) No clutter, $\mathbf{R}_k = 4 \cdot \mathbf{I}\,\mathrm{m}^2$.

(b) With clutter, $\mathbf{R}_k = 16 \cdot \mathbf{I}\,\mathrm{m}^2$.

Fig. 8: Results of evaluation with static target. Light red are example positive measurements, negative measurements omitted. Clutter parameters are $\mathrm{p}_{FN} = \mathrm{p}_{FP} = 0.1$.

In Fig. 8a, it can be seen that the NM estimate is smaller (or *tighter*) than the RHM. This is explained by the fact that negative measurements push the estimate towards the inside, an effect that is absent in RHMs. Still, both approximations can be seen to be accurate. However, when clutter is present and the measurement noise is higher (Fig. 8b), the RHM starts to diverge. As a reminder, both approaches use the same gating procedure. The problem is that the clutter measurements which pass the validation gate push the boundary towards the outside, an artifact which accumulates in time. The proposed model compensates for this with the use of negative measurements, which push the boundary to its correct form.

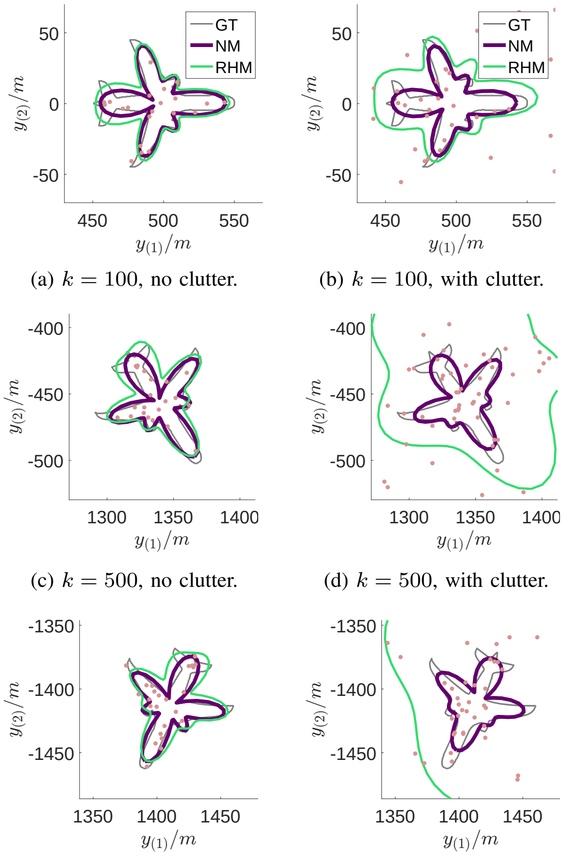
B. Moving Target

For this experiment, the target is moving at a constant speed of 5 m per timestep, where each timestep represents a time of $\Delta t = 0.01\,\mathrm{s}$. From timestep 0 to 100, the plane is moving from $\left[0, 0\right]^T$ to $\left[0, 500\right]^T$ in a straight line. Then, the plane begins rotating southwards around the point $\left[500, -1000\right]^T$. The initial position for both estimators is $\left[0, 0\right]^T$, and the speed is set to the correct value of $500\,\mathrm{m}\,/\,\mathrm{s}$. As with the static target, the initial shape is a circle of radius 30 m. Fig. 9 shows the results of selected timesteps averaged over 50 runs.

At the end of the straight line motion (Fig. 9a, Fig. 9b), both estimates are still accurate. In fact, assuming no clutter and low noise (Fig. 9c, Fig. 9e), the RHM and the proposed approach return similar results. However, once clutter is present and noise is higher, the RHM cannot keep up with the target. As seen in Fig. 9d and Fig. 9f, the estimator begins to diverge completely, so that by the end of the arc the estimate is larger than the sensor field of view. The proposed approach, however, is able to keep up by incorporating additional information from negative measurements.

VI. Conclusion

While traditional techniques to track extended targets exploit only positive measurements, which are assumed to stem from the target, in this paper we presented an approach that also incorporates negative measurements, which tell us where the target is *not*. We developed a recursive Bayesian estimator that represents the target as a star-convex shape, and draws from ideas of curve fitting and distance minimization techniques. Two alternative implementations were provided, one for LRKFs such as the well-known UKF, and another for particle filters. The evaluation showed that, in conditions with high measurement quality, the results were very similar to state-of-the-art approaches such as RHMs. However, in the presence of clutter and high measurement noise, the RHM diverged while the proposed approach was still able to return accurate results, thus demonstrating the advantages of extracting more information from negative measurements.



(a) $k = 100$, no clutter.

(b) $k = 100$, with clutter.

(c) $k = 500$, no clutter.

(d) $k = 500$, with clutter.

(e) $k = 800$, no clutter.

(f) $k = 800$, with clutter.

Fig. 9: Results of evaluation with moving target. Left column uses $\mathbf{R}_k = 4 \cdot \mathbf{I}\,\mathrm{m}^2$, right column $\mathbf{R}_k = 8 \cdot \mathbf{I}\,\mathrm{m}^2$. Clutter parameters are $\mathrm{p}_{FN} = \mathrm{p}_{FP} = 0.1$. Light red are example positive measurements, negative measurements are omitted.

References

[1] M. Werman and D. Keren, “A Bayesian method for fitting parametric and nonparametric models to noisy data,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 5, pp. 528–534, 2001.

[2] F. Faion, A. Zea, M. Baum, , and U. D. Hanebeck, “Bayesian Estimation of Line Segments,” in *Proceedings of the IEEE ISIF Workshop on Sensor Data Fusion: Trends, Solutions, Applications (SDF 2014)*, Bonn, Germany, Oct. 2014.

[3] C. Lundquist, U. Orguner, F. Gustafsson, and S. Member, “Extended target tracking using polynomials with applications to road-map estimation,” *IEEE Transactions on Signal Processing*, pp. 1–12, 2011.

[4] M. Feldmann, D. Franken, and W. Koch, “Tracking of extended objects and group targets using random matrices,” *Signal Processing, IEEE Transactions on*, vol. 59, no. 4, pp. 1409–1420, 2011.

[5] J. Lan and X. R. Li, “Tracking of maneuvering non-ellipsoidal extended object or target group using random matrix,” *IEEE Transactions on Signal Processing*, vol. 62, no. 9, pp. 2450–2463, May 2014.

[6] K. Granstrom, P. Willett, and Y. Bar-Shalom, “An extended target tracking model with multiple random matrices and unified kinematics,” in *Information Fusion (Fusion), 2015 18th International Conference on*, July 2015, pp. 1007–1014.

[7] M. Baum and U. D. Hanebeck, “Shape Tracking of Extended Objects and Group Targets with Star-Convex RHMs,” in *Proceedings of the 14th International Conference on Information Fusion (Fusion 2011)*, Chicago, Illinois, USA, 2011.

[8] L. Sun, X. Li, and J. Lan, “Extended object tracking based on support functions and extended gaussian images,” in *Information Fusion (FUSION), 2013 16th International Conference on*, July 2013.

[9] N. Wahlstrom and E. Ozkan, “Extended target tracking using gaussian processes,” *Signal Processing, IEEE Transactions on*, vol. 63, no. 16, pp. 4165–4178, 2015.

[10] P. Lancaster and K. Salkauskas, “Curve and surface fitting. an introduction,” *London: Academic Press, 1986*, vol. 1, 1986.

[11] P. S. Bandyopadhayay, R. J. Boik, and P. Basu, “The curve fitting problem: A bayesian approach,” *Philosophy of Science*, vol. 63, pp. S264–S272, 1996.

[12] M. Baum, B. Noack, and U. D. Hanebeck, “Extended Object and Group Tracking with Elliptic Random Hypersurface Models,” in *Proceedings of the 13th International Conference on Information Fusion (Fusion 2010)*, Edinburgh, United Kingdom, Jul. 2010.

[13] F. Faion, A. Zea, M. Baum, and U. D. Hanebeck, “Partial Likelihood for Unbiased Extended Object Tracking,” in *Proceedings of the 18th International Conference on Information Fusion (Fusion 2015)*, Washington D. C., USA, Jul. 2015.

[14] W. Koch, “On ’negative’ information in tracking and sensor data fusion: Discussion of selected examples,” in *Proceedings of the Seventh International Conference on Information Fusion*, vol. 1. IEEE Publ. Piscataway, NJ, 2004, pp. 91–98.

[15] ——, “On exploiting ’negative’ sensor evidence for target tracking and sensor data fusion,” *Information Fusion*, vol. 8, no. 1, 2007.

[16] J. Hoffman, M. Spranger, D. Gohring, and M. Jungel, “Making use of what you don’t see: negative information in markov localization,” in *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, Aug 2005, pp. 2947–2952.

[17] A. Zea, F. Faion, and U. D. Hanebeck, “Exploiting Clutter: Negative Information for Enhanced Extended Object Tracking,” in *Proceedings of the 18th International Conference on Information Fusion (Fusion 2015)*, Washington D. C., USA, Jul. 2015.

[18] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking,” *IEEE Transactions on Signal Processing*, vol. 50, no. 2, Feb 2002.

[19] S. J. Julier and J. K. Uhlmann, “Unscented Filtering and Nonlinear Estimation,” in *Proceedings of the IEEE*, vol. 92, no. 3, 2004.

[20] K. Gilholm and D. Salmond, “Spatial Distribution Model for Tracking Extended Objects,” *IEE Proceedings on Radar, Sonar and Navigation*, vol. 152, no. 5, pp. 364–371, 2005.

[21] J. Steinbring and U. D. Hanebeck, “S2KF: The Smart Sampling Kalman Filter,” in *Proceedings of the 16th International Conference on Information Fusion (Fusion 2013)*, Istanbul, Turkey, Jul. 2013.

[22] K. Gilholm, S. Godsill, S. Maskell, and D. Salmond, “Poisson Models for Extended Target and Group Tracking,” in *SPIE: Signal and Data Processing of Small Targets*, 2005.

[23] J. Steinbring and U. D. Hanebeck, “Progressive Gaussian Filtering Using Explicit Likelihoods,” in *Proceedings of the 17th International Conference on Information Fusion (Fusion 2014)*, Salamanca, Spain, Jul. 2014.

[24] Y. Bar-Shalom and T. Fortmann, *Tracking and Data Association*, ser. Mathematics in Science and Engineering Series. Academic Press, 1988.