Level-Set Random Hypersurface Models for Tracking Nonconvex Extended Objects

ANTONIO ZEA
FLORIAN FAION
Karlsruhe Institute of Technology (KIT)
Karlsruhe, Germany

MARCUS BAUM
University of Göttingen
Göttingen, Germany

UWE D. HANEBECK
Karlsruhe Institute of Technology (KIT)
Karlsruhe, Germany

This paper presents a novel approach to track a nonconvex shape approximation of an extended target based on noisy point measurements. For this purpose, a novel type of random hypersurface model (RHM) called Level-set RHM is introduced that models the interior of a shape with level-sets of an implicit function. Based on the Level-set RHM, a nonlinear measurement equation can be derived that allows to employ a standard Gaussian state estimator for tracking an extended object even in scenarios with moderate measurement noise. In this paper, shapes are described using polygons, and shape regularization is applied using ideas from active contour models.

Manuscript received November 6, 2013; revised November 15, 2014, July 29, 2015; released for publication May 25, 2016.

DOI. No. 10.1109/TAES.2016.130704.

Refereeing of this contribution was handled by W. Koch.

This is an extended version of the paper, “Level-Set Random Hypersurface Models for Tracking Non-Convex Extended Objects,” published at the *16th International Conference on Information Fusion (Fusion 2013)*. Significant changes have been made with regard to the derivation of the probabilistic model and the practical evaluation.

Authors’ address: A. Zea, F. Faion, U. D. Hanebeck, Karlsruhe Institute of Technology, Institute for Anthropomatics and Robotics, Adenauerring 2, Gebäude 50.20, Karlsruhe, 76131, Germany. M. Baum, Goldschmidtstr. 7, 37077 Göttingen, Germany. Corresponding author is A. Zea, E-mail: (antonio.zea@kit.edu).

I. INTRODUCTION

In target tracking scenarios [1], the usual conditions are high noise and only a few measurements per timestep. As a consequence, common tracking algorithms have been designed to model the target object as a point with no extent. Nonetheless, as sensor quality increases and more measurements become available, this assumption no longer holds. A better approach to increase robustness and accuracy is to explicitly model the target extent and consider it in the algorithm.

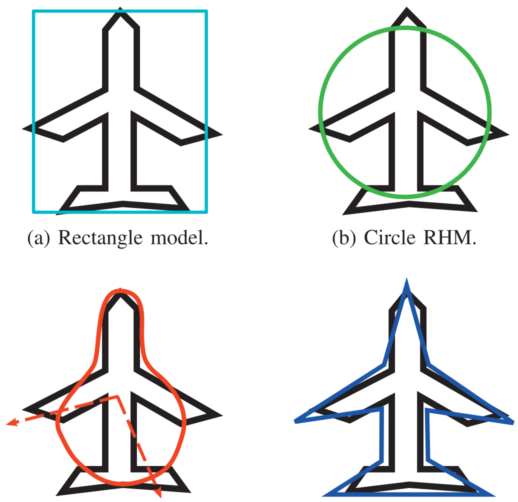
Widely used shape approximations are in the form of ellipses [2–4], sticks [5, 6], or rectangles [7, 8] [Fig. 1(a)]. Other approaches estimate shapes using Gaussian images [9] and border parameterization [10]. However, in many applications, measurements do not only originate from the object boundary, but also from the interior of the target. For these scenarios, standard fitting algorithms cannot be applied [11]. In addition, shape tracking algorithms usually do not consider scenarios where measurements have individual, nonisotropic noise parameters, which arise in the context of, e.g., sensor fusion [12]. Altogether, an explicit model is needed that captures this wide range of measurement characteristics.

Work has been done to model such scenarios in the form of random hypersurface models (RHMs) [13]. Representations include approximations using circular [Fig. 1(b)] and elliptical shapes, best suited for scenarios with very low quality measurements, when high sensor noise makes it almost impossible to obtain further details. For scenarios with higher measurement quality, more detail can be captured using star-convex shape models [Fig. 1(c)]. This paper builds upon the theoretic concepts of these works and presents a generalization for arbitrary, simply connected shapes.

Besides the RHM approach to extended object tracking [13–16], there are essentially two lines of work in literature. First, particle filter methods [5, 17–19] have been proposed for dealing with the nonlinearity of the estimation problem. Second, random matrix theory [20–25] allows to derive analytic expressions for targets that are modeled as ellipses.

The main contribution is a generalized RHM that uses level-sets for modeling targets [Fig. 1(d)]. In particular, in this paper we are interested in targets with filled shapes, i.e., where measurements are also generated from the interior of the target. To the best of our knowledge, this is the first extended object tracking method for explicitly estimating detailed arbitrary nonconvex, and particularly non-star-convex, geometric objects.

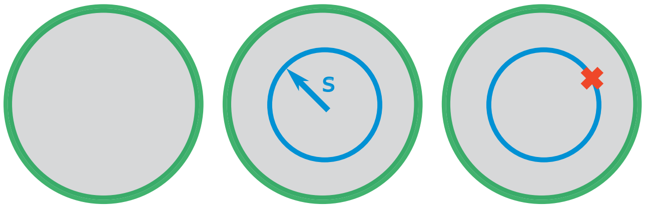
The remainder of this paper is structured as follows. First, the general theoretical concepts of the models involved in extended object tracking are outlined in Section II. In Section III, the new Level-set RHMs are introduced and explained in detail. An implementation of Level-set RHMs using a polygonal shape representation is described in Section IV, and a more detailed discussion in Section V. In Section VI, the proposed approach is



(a) Rectangle model. (b) Circle RHM.

(c) Star-convex RHM. (d) Level-set RHM.

Fig. 1. Representation models for extended objects.



(a) (b) (c)

Figure 2: Random Hypersurface Model for a disk.

Fig. 2. RHM for disk.

demonstrated and evaluated using synthetic data and real-life data from an RGBD sensor, which captures both color (RGB) and depth information from a scene. Finally, this work concludes with a short summary and an outlook to future work in Section VII.

II. MODELING EXTENDED OBJECTS

In this section, a probabilistic model for an extended object to be tracked is described. This includes appropriate models to specify the state to be estimated, the target shape, the measurement generation process, and the temporal evolution of the target.

A. Shape Model

This work focuses on tracking compact, simply connected shapes. In other words, we are concerned with closed shapes of finite size that may be filled, but which do not contain holes. The parameters of the target shape are represented with the parameter vector $\underline{x}^p$. Then, the set $\mathcal{S}(\underline{x}^p)$ denotes the shape itself, including its interior. As the shape may change in time, the parameters at the discrete timestep k are written as $\underline{x}_k^p$, and the shape as $\mathcal{S}(\underline{x}_k^p)$.

Given that the shape will be stochastically associated with noisy measurements, the uncertain knowledge about the shape parameters is represented as the random vector $\underline{\boldsymbol{x}}_k^p$. Note that the state vector does not have to be limited to the shape parameters only. Thus, the state vector $\underline{\boldsymbol{x}}_k$ is assumed to contain $\underline{\boldsymbol{x}}_k^p$, and additional parameters as required by further models, such as a motion model or others.

B. Measurement Model

At each timestep k, a set of point measurements $\mathcal{Y}_k = \{\underline{y}_{k,1}, \dots, \underline{y}_{k,l}\}$ becomes available. Each measurement $\underline{y}_{k,i}$ represents a position in Cartesian coordinates, i.e., $\underline{y}_{k,i} \in \mathbb{R}^d$, where d is usually two or three. It is not assumed that the number of measurements l carries information about $\underline{x}_k$. The index i will be dropped for legibility unless needed.

It is assumed that each measurement $\underline{y}_k$ was generated by a source $\underline{z}_k$ that belongs to the shape $\mathcal{S}(\underline{x}_k^p)$. However, $\underline{y}_k$ and $\underline{z}_k$ are generally not equal, as observation by a sensor is assumed to introduce noise. In this paper, this sensor noise is assumed to be additive, zero-mean, and Gaussian distributed, so that the relation between $\underline{y}_k$ and $\underline{z}_k$ can be described as

$$\underline{y}_k = \underline{z}_k + \underline{w}_k \ , \tag{1}$$

where $\underline{w}_k$ is denoted as the noise term, assumed to be a realization of the random vector $\underline{\boldsymbol{w}}_k \sim \mathcal{N}(\underline{0}, \mathbf{C}_k^w)$. The covariance matrix $\mathbf{C}_k^w$ is determined by the sensor model. The measurement noise term $\underline{w}_k$ is assumed to be stochastically independent from other noise terms and from the state.

C. Dynamic Model

The dynamic model describes how the state $\underline{\boldsymbol{x}}_k$ evolves in time between successive timesteps k and $k+1$. A motion model, for example, assumes that a position will change in time following a velocity or acceleration parameter. The system function describes this process, and has the form

$$\underline{\boldsymbol{x}}_{k+1} = a_k(\underline{\boldsymbol{x}}_k, \underline{\boldsymbol{r}}_k) \, , \tag{2}$$

where $\underline{\boldsymbol{r}}_k$ represents the process noise.

III. LEVEL-SET RHMS

In this section we introduce Level-set RHMs. First, as background, we present the concept of shape functions, which allows us to describe shapes without needing an explicit generative model for sources. Then, we explain the ideas behind RHMs. Finally, we describe our contribution in detail.

A. Shape Function

When estimating shape parameters, a commonly used concept is the so-called shape functions, which return for a point $\underline{\tilde{z}}$ a scalar that determines how well it fits a given shape. More formally, let there be a given shape $\mathcal{S}(\underline{x}_k^p)$. We say that the continuous function $\phi(\underline{x}_k^p, \underline{\tilde{z}})$ is a shape function of $\mathcal{S}(\underline{x}_k^p)$ iff it holds that

$$\mathcal{S}\left(\underline{x}_k^p\right) = \{\underline{\tilde{z}} \in \mathbb{R}^d \mid \phi(\underline{x}_k^p, \underline{\tilde{z}}) = 0\} \ , \tag{3}$$

i.e., only points that are part of the shape can return a value of 0. A widely used shape function is the Euclidian distance function [26]. It can be generalized as the Mahalanobis distance function, which, for a given positive-definite matrix $\mathbf{\Sigma}$, is defined as

$$\phi^M(\underline{x}_k^p, \underline{\tilde{z}}) = \min_{\underline{z}_k \in \mathcal{S}(\underline{x}_k^p)} \sqrt{(\underline{\tilde{z}} - \underline{z}_k)^T \cdot \mathbf{\Sigma}^{-1} \cdot (\underline{\tilde{z}} - \underline{z}_k)} \ , \quad (4)$$

i.e., the smallest Mahalanobis distance to a point in $\mathcal{S}(\underline{x}_k^p)$. Note that the Mahalanobis distance is equivalent to the Euclidian distance for $\mathbf{\Sigma} = \mathbf{I}$.

A usual extension to shape functions, and the focus of this paper, are signed shape functions, identical in function but which return a negative value if $\underline{\tilde{z}}$ is outside the shape.[1] These include the signed Euclidian and Mahalanobis distance functions. An example follows.

Example 1 (Shape Function of a Circle)

Let $\mathcal{S}(\underline{x}_k^p)$ be the dark green circle in Fig. 2(a), centered in the origin, and with the only shape parameter $\underline{x}_k^p$ being the radius r_k. In addition, let $\underline{y}_k$ be a measurement. Then, we can design the signed Euclidian distance function in the form of

$$\phi^c(\underline{x}_k^p, \underline{\tilde{z}}) = r_k - ||\underline{\tilde{z}}|| \ , \quad (5)$$

where $|| \cdot ||$ is the Euclidian norm. It can be seen that $\phi^c(\underline{x}_k^p, \underline{\tilde{z}})$ is positive if $\underline{\tilde{z}}$ is inside the circle, 0 if $\underline{\tilde{z}}$ belongs to $\mathcal{S}(\underline{x}_k^p)$, and negative if outside.

Using a shape function and (1), we can derive easily a measurement function for a measurement $\underline{y}_k$ with source $\underline{z}_k$ in the form of

$$\begin{aligned} h(\underline{x}_k, \underline{y}_k, \underline{w}_k) &:= \phi(\underline{x}_k^p, \underline{z}_k) \\ &= \phi(\underline{x}_k^p, \underline{y}_k - \underline{w}_k) \ , \end{aligned} \quad (6)$$

leading to the measurement equation

$$h(\underline{x}_k, \underline{y}_k, \underline{w}_k) = 0 \ . \quad (7)$$

This is an implicit measurement equation, where the target state, the measurement, and the measurement noise are propagated through $h(\cdot, \cdot, \cdot)$ and then associated with the pseudo-measurement 0. Thus, informally, it can be said that the objective is for the filter to estimate $\underline{x}_k$ so that $h(\cdot, \cdot, \cdot)$ is as close to 0 as possible. Note that the unknown source $\underline{z}_k$ is not probabilistically modeled, and instead only described through the relationship in (1). This approach, particularly popular in the field of shape fitting, allows for simple and fast estimation. In particular, this avoids the requirement of an explicit generative model for the selection of a source (such as in [5]), which can become extremely complex for filled nonconvex shapes. However, this approach implicitly assumes that the source is the *closest* point to the measurement, according to the used metric such as (4). This assumption is appropriate for low noise levels, as the closest source is a good approximation of the true source. For higher noise levels, however, a mechanism to compensate this problem is needed, as the assumption of an incorrect source causes estimation bias. This challenge is discussed in Section V-D, and a more detailed treatment can be found in [27].

Note that in this paper we are concerned with filled shapes. For these, a measurement function such as (6) is not helpful, as shown in [28, 29]. For visualization, let us describe the following thought experiment where the disk in Fig. 2(a) is the target. As $\mathcal{S}(\underline{x}_k^p)$ is a filled target, the shape function in the interior also returns 0. In this imaginary case, the estimator is initialized with a much larger disk which completely contains the target. Then, after receiving measurements from the target, it turns out that they are all in the interior of the oversized estimate, and thus, $\phi(\underline{x}_k^p, \underline{z}_k)$ will always return the desired value of 0. In consequence, the estimator cannot recognize that the estimate is too large, and remains stuck with the incorrect result. Section III-B introduces an approach to deal with these filled shapes.

B. RHMs

An RHM [13] is a probabilistic model to describe how to generate single measurements from an extended shape. The key idea is to interpret a target shape as the transformation of an underlying shape, such as its boundary, using a random variable $\boldsymbol{s}_k$. This process can be visualized by describing the model for generating a measurement. First, a transformed version of the underlying shape is produced by randomly drawing a transformation factor $\boldsymbol{s}_k$ from $\boldsymbol{s}_k$. Then, the measurement source is selected from this transformed shape. Finally, the measurement source is corrupted during observation by noise as described by (1). Note that, as the transformed shape is described with a shape function, we do not model how the source is selected from its transformed shape.

The transformation can interpret the factor $\boldsymbol{s}_k$ in any way [7, 14, 16], depending on the model, and does not need to be linear. The distribution of $\boldsymbol{s}_k$. represents how probable it is that each transformed shape generates a source, and is assumed to be known a priori. In addition, it is assumed that the transformation factors are stochastically independent from each other and from the state. An example with a circular disk follows.

Example 2 (RHM with a circular disk)

Let $\mathcal{S}(\underline{x}_k^p)$ be the disk in Fig. 2*(b), i.e., the dark green circle and its interior, centered in the origin, and with the only shape parameter $\underline{x}_k^p$ being the radius r_k. In this example, we describe $\mathcal{S}(\underline{x}_k^p)$ as an RHM by transforming its boundary (dark green circle), by using the transformation parameter $\boldsymbol{s}_k \sim \mathcal{U}(0, 1)$.*

We model an observed measurement as being the result of the following process. First, a factor s_k drawn from $\mathbf{S}_k$ scales the circular boundary homogeneously, as seen in Fig. 2*(b), producing a circle of radius $\boldsymbol{s}_k \cdot r_k$. Then, the measurement source $\underline{z}_k$ is selected from this scaled*

[1] Many implementations of signed shape functions use the negative sign for points inside the shape. However, adapting them to the convention used in this paper is only a matter of flipping the sign.

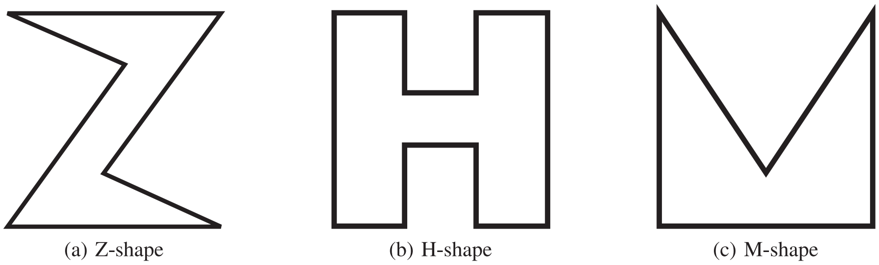

(a) Z-shape (b) H-shape (c) M-shape

Fig. 3. Selection of shapes.

circle [Fig. 2(c)]. *Finally, the source is corrupted by sensor noise as described in* (1), *producing the measurement* $\underline{y}_k$. *Note that we are not interested in how the source* $\underline{z}_k$ *is selected from the scaled circle.*

In a similar fashion as (5), *we obtain the shape function for the transformed shape*

$$\phi^s(\underline{x}_k^p, \underline{\tilde{z}}) = s_k \cdot r_k - ||\underline{\tilde{z}}|| \,. \tag{8}$$

Using this shape function, we can derive a measurement equation. As in (6), *we fuse* (1) *and* (8) *to obtain the following measurement function*

$$\begin{aligned} h(\underline{x}_k, \underline{y}_k, \underline{w}_k, s_k) =& \ \phi^s(\underline{x}_k^p, \underline{z}_k) \\ =& \ s_k \cdot r_k - ||\underline{z}_k|| \\ :=& \ s_k \cdot r_k - ||\underline{y}_k - \underline{w}_k|| \,, \end{aligned} \tag{9}$$

leading to the measurement equation

$$h(\underline{x}_k, \underline{y}_k, \underline{w}_k, s_k) = 0 \,. \tag{10}$$

Note that, because of the measurement noise, the exact s_k *used to generate* $\underline{y}_k$ *is not known. In consequence, we interpret* s_k *as an additional noise term.*

RHMs can be contrasted to the approach in Section III-A, in that RHMs model the selection of the transformed shape explicitly through s_k, but leave the selection of the source from the transformed shape implicit.

C. Level-Set RHMs

While a transformation mechanism for circles is simple, i.e., scaling the radius, for more complex shapes a way to implement the transformation is not straightforward. Fig. 3 shows examples of these shapes. A circle is an example of a convex shape. A shape $\mathcal{S}(\underline{x}_k^p)$ is called convex if, for every pair of points that belong to it, the segment joining both points is completely inside $\mathcal{S}(\underline{x}_k^p)$. Shapes such as the M-shape [Fig. 3(c)] are clearly not convex, however, they are star-convex. The defining property of a star-convex shape is that there exists at least a point $\underline{m}$ of $\mathcal{S}(\underline{x}_k^p)$, so that, for every point in $\mathcal{S}(\underline{x}_k^p)$, the segment connecting it with $\underline{m}$ is completely contained in $\mathcal{S}(\underline{x}_k^p)$. This does not apply to shapes like the Z-shape [Fig. 3(a)] or the H-shape [Fig. 3(b)], as no such point exists for them. Thus, while the interior of a star-convex shape can be described by shrinking its boundary towards $\underline{m}$, as applied in [16], this cannot work for shapes like those in Fig. 3(a).

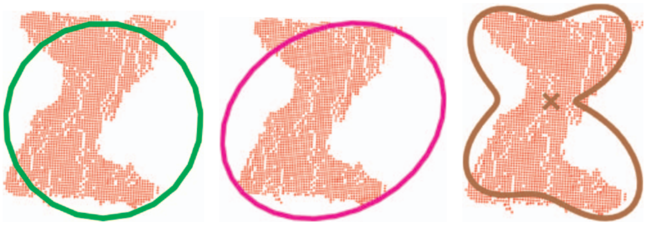

(a) Circle RHM (b) Ellipse RHM (c) Star-convex RHM

Fig. 4. Z-shape object tracked using different RHMs.

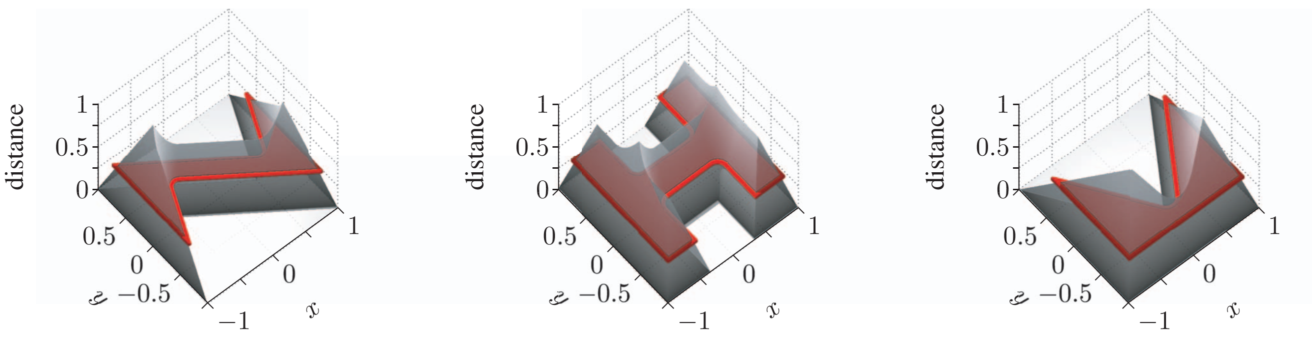
Fig. 4 shows a real-life Z-shape as observed by an RGBD sensor, and then estimated using different RHMs. Fig. 4(a) is tracked using a circle RHM [30], while Fig. 4(b) uses an elliptic RHM [31]. Both are suitable for tracking the basic size of the object, but cannot estimate any further details. For Fig. 4(c) a star-convex RHM [16] is used, which is capable of estimating a much more detailed form, as long as the target shape is star-convex. However, given that the Z-shape does not fulfill this condition, the estimate is not close to the target. This paper aims to work upon these established concepts in order to track arbitrary, nonconvex shapes.

The main task to develop a more general RHM is to find a suitable and general transformation mechanism for arbitrary filled shapes. As described in Section III-A, the shape function of a filled shape is not useful for describing its interior, where all points have a value of 0. Instead, let us focus on the shape function of its boundary $\phi^b(\underline{x}_k^p, \underline{\tilde{z}})$. Thus, for the Level-set RHMs that are introduced in this paper, shape transformation is implemented using the concepts of level-sets of the boundary shape functions.

Definition 1 (Level-set) *For a given shape function* $\phi^b(\underline{x}_k^p, \underline{\tilde{z}})$, *a level-set* $\mathcal{L}_\phi(c)$ *is the region where the shape function takes the value of c, i.e.,*

$$\mathcal{L}_\phi(c) := \{\underline{\tilde{z}} \in \mathbb{R}^d \mid \phi^b(\underline{x}_k^p, \underline{\tilde{z}}) = c\}. \tag{11}$$

It can be seen that every point $\underline{\tilde{z}}$ belongs to a level-set. In addition, the target shape boundary is by definition $\mathcal{L}_\phi(0)$. Fig. 5 shows the level-sets of the shapes in Fig. 3 using the signed Euclidian distance (implemented in Section IV-B), with the highlighted level-set $\mathcal{L}_\phi(0.3)$. The explicit calculation of level-sets is not straightforward, and a



(a) Z-shape

(b) H-shape

(c) M-shape

Fig. 5. Level-sets for different shapes in Fig. 3. Shape function in gray, level-set $\mathcal{L}_\phi(0.3)$ in red.

variety of modeling techniques exist [26, 32, 33]. However, Level-set RHMs do not require an explicit calculation for level-sets.

It makes sense to assume that the target shape can be interpreted as a closed set of points. From this assumption, it follows that $\phi^b(\underline{x}_k^p, \underline{\tilde{z}})$ must have a maximum, as only the bounded interior of the shape returns positive values. Then, we proceed to define $\phi_{max}(\underline{x}_k^p)$ as

$$\phi_{max}(\underline{x}_k^p) := \max_{\underline{\tilde{z}} \in \mathcal{S}(\underline{x}_k^p)} \phi^b(\underline{x}_k^p, \underline{\tilde{z}}) \, . \tag{12}$$

In particular, $\mathcal{S}(\underline{x}_k^p)$ can be seen as the union of all level-sets $\mathcal{L}_\phi(c)$ for $c \in [0, \phi_{max}(\underline{x}_k^p)]$.

From this, a generative model for all points in $\mathcal{S}(\underline{x}_k^p)$ can be derived in a similar fashion as Example 2, using level-sets as the transformation. The distribution of the transformation parameter s_k can be arbitrary, but its support is assumed to be [0, 1]. We model an observed measurement as being the result of the following process. First, a transformation factor s_k is drawn from $\boldsymbol{s}_k$, producing the level-set $\mathcal{L}_\phi(s_k \cdot \phi_{max}(\underline{x}_k^p))$. Then, the measurement source $\underline{z}_k$ is selected from this level-set. As mentioned before, the mechanism of this selection is not modeled explicitly. Then, as $\underline{z}_k$ belongs to $\mathcal{L}_\phi(s_k \cdot \phi_{max}(\underline{x}_k^p))$, by definition it follows that

$$\phi^b(\underline{x}_k^p, \underline{\tilde{z}}_k) = s_k \cdot \; \phi_{max}(\underline{x}_k^p) \, . \tag{13}$$

Finally, the source is corrupted by sensor noise as described in (1), producing the measurement $\underline{y}_k$.

Using (13), we can derive a shape function for the transformed shape in the form of

$$\phi^s(\underline{x}_k^p, \underline{\tilde{z}}) = s_k \, \cdot \phi_{max}(\underline{x}_k^p) - \phi^b(\underline{x}_k^p, \underline{\tilde{z}}) . \tag{14}$$

Finally, based on (1) and (14), and using ideas from example 2, the following measurement function for a measurement $\underline{y}_k$ can be obtained in the form of

$$\begin{aligned} h(\underline{x}_k, \underline{y}_k, \; \underline{w}_k \; , \; \underline{s}_k) &:= \phi^s(\underline{x}_k^p, \underline{z}_k) \\ &= \; \phi^s(\underline{x}_k^p, \underline{y}_k - \underline{w}_k) \\ &= \; s_k \cdot \; \phi_{max}(\underline{x}_k^p) - \phi^b(\underline{x}_k^p, \underline{y}_k - \underline{w}_k) \, , \end{aligned} \tag{15}$$

leading to the measurement equation

$$h(\underline{x}_k, \underline{y}_k, \underline{w}_k, \underline{s}_k) = 0 \, . \tag{16}$$

Similarly, as the s_k that generated $\underline{z}_k$ is not known, we interpret s_k as an additional noise term.

IV. Level-set RHMs using Polygons

In this section, a Level-set RHM is implemented using polygons to represent a wide variety of two-dimensional shapes. The key properties of polygons are their simple mathematical representation, great flexibility for describing shapes, ease of implementation of a shape function, and, as will be seen in Section V-C, they allow for a straightforward approach for regularization. In addition, polygons are a widely researched topic in fields such as computer graphics and computer vision, producing a wide background in literature for possible model extensions.

A. Polygons

Let $\{\underline{b}_{k,0}, \ldots, \underline{b}_{k,n-1}\}$ be a sequence of n points in $\mathbb{R}^2$. The curve formed by the connected segments between consecutive points is called a polygonal chain. This papers deals with closed polygonal chains, i.e., where $\underline{b}_{k,n-1}$ is also connected to $\underline{b}_{k,0}$. The points in the chain sequence are referred as the vertices of the polygon.

The polygon parameters can be expressed with a parameter vector. The representation used in this paper is

$$\underline{x}_k^p = \left[\underline{b}_{k,0}^T, \ldots, \underline{b}_{k,n-1}^T \right]^T \, , \tag{17}$$

where each point $\underline{b}_{k,j}$ is represented in absolute Cartesian coordinates. For simplicity, the index of a polygon point is assumed to lie in $\mathbb{Z}/n\mathbb{Z}$, i.e., to wrap around the interval $[0, n-1]$. Thus, $\underline{b}_{k,n}$ is equivalent to $\underline{b}_{k,0}$, and $\underline{b}_{k,-1}$ is equivalent to $\underline{b}_{k,n-1}$. In this section, $\mathcal{S}(\underline{x}_k^p)$ is used to describe the filled polygon, i.e., the segments and the interior area.

Note that the number of polygon vertices is assumed to be known a priori. While approaches can be developed for dynamically estimating the number of vertices to best describe the target, we consider this topic to be out of the scope of this work.

B. Boundary Shape Function

The boundary shape function to be used in polygonal RHMs is the signed Mahalanobis distance, denoted as $\phi^{\pi}(\underline{x}_k^p, \underline{\tilde{z}})$. In order to describe it, first we need a function $d(\underline{x}_k^p, \underline{\tilde{z}})$ that returns the Mahalanobis distance between an arbitrary point $\underline{\tilde{z}}$ and the closest point in the boundary of the polygon $\mathcal{S}(\underline{x}_k^p)$. Usual choices for $\mathbf{\Sigma}$ are either $\mathbf{I}$, in case of isotropic noise or when speed is needed, or $\boldsymbol{C}_k^w$ for the general case, especially when nonisotropic noise is present.

A simple implementation is as follows. Let (u) be a function that clamps $u \in \mathbb{R}$ to the interval $[0, 1]$, defined as

$$\text{clamp}(u) = \begin{cases} 0, \text{ if } u < 0 \\ u, \text{ if } 0 \leq u \leq 1 \\ 1, \text{ otherwise.} \end{cases} \tag{18}$$

Then, for each segment j connecting $\underline{b}_{k,j}$ and $\underline{b}_{k,j+1}$, the point $\underline{z}_j^{\pi}$ in the polygon that is closest to the given point $\underline{\tilde{z}}$ can be determined with

$$u_j := \frac{(\underline{\tilde{z}} - \underline{b}_{k,j})^T \cdot \mathbf{\Sigma}^{-1} \cdot (\underline{b}_{k,j+1} - \underline{b}_{k,j})}{(\underline{b}_{k,j+1} - \underline{b}_{k,j})^T \cdot \mathbf{\Sigma}^{-1} \cdot (\underline{b}_{k,j+1} - \underline{b}_{k,j})} \tag{19}$$

and

$$\underline{z}_j^{\pi} = \underline{b}_{k,j} + \text{clamp}(u_j) \cdot \left(\underline{b}_{k,j+1} - \underline{b}_{k,j}\right). \tag{20}$$

Finally, the minimum Mahalanobis distance between the polygon and $\underline{\tilde{z}}$ is simply

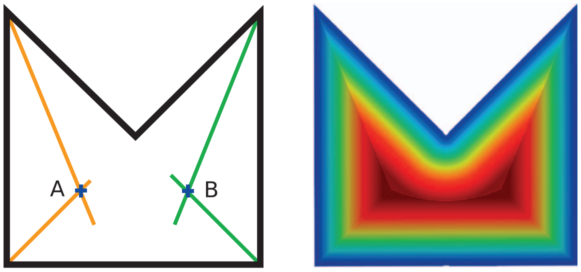
$$d(\underline{x}_k^p, \underline{\tilde{z}}) = \min_{0 \leq j \leq n-1} \sqrt{(\underline{\tilde{z}} - \underline{z}_j^{\pi})^T \cdot \mathbf{\Sigma}^{-1} \cdot (\underline{\tilde{z}} - \underline{z}_j^{\pi})}. \tag{21}$$

Thus, the complexity of the distance algorithm is in $\mathcal{O}(n)$. The signed distance function follows from this, in the form of

$$\phi^{\pi}(\underline{x}_k^p, \underline{\tilde{z}}) := \begin{cases} d(\underline{x}_k^p, \underline{\tilde{z}}), & \text{if } \underline{\tilde{z}} \in \mathcal{S}\left(\underline{x}_k^p\right) \\ -d(\underline{x}_k^p, \underline{\tilde{z}}), & \text{otherwise.} \end{cases} \tag{22}$$

Checking if $\underline{\tilde{z}}$ is inside $\mathcal{S}(\underline{x}_k^p)$ can be done using, e.g., the odd-even rule or winding numbers [34]. These algorithms are usually in $\mathcal{O}(n)$.

The only remaining piece is the calculation of $\phi_{max}^{\pi}(\underline{x}_k^p)$, i.e., the maximum value of the shape function for $\underline{x}_k^p$. For polygons, there are usually many points whose shape function evaluates to the maximum value. However, we can exploit the fact that the set $\mathcal{Z}^I$ of intersections of vertex angle bisectors will always contain at least one of those points. In other words, in order to find $\phi_{max}^{\pi}(\underline{x}_k^p)$, it is sufficient to find the maximum value of $\phi^{\pi}(\underline{x}_k^p, \underline{z}^I)$ from all intersections $\underline{z}^I \in \mathcal{Z}^I$. Fig. 6 shows an example of this procedure. For the M-shape, four selected vertex angle bisectors are shown in colored lines. From these bisectors, the intersections A and B [Fig. 6(a)] yield the maximum distances from $\mathcal{Z}^I$. The other bisector intersections lie outside the shape, and thus, yield negative distances. In Fig. 6(b), the distance function for points inside of the shape is plotted. It can be verified that points A and B indeed correspond to the maximum distance $\phi_{max}^{\pi}(\underline{x}_k^p)$. This algorithm yields a complexity of $\mathcal{O}(n^3)$.



(a) Polygon bisectors. (b) Distance function.

Fig. 6. Calculation of $\phi_{max}^{\pi}(\underline{x}_k^p)$ using bisectors.

V. IMPLEMENTATION DISCUSSION

The task of this work is to track arbitrary, nonconvex extended objects using level-sets. This raises, however, the following issues. First, it is difficult to determine how the tracking algorithm should be initialized. Second, the probabilistic approach used in RHMs requires the distribution for the transformation parameter s_k to be known beforehand. Third, when prior information about the tracked object and the quality of the measurements is sparse, a regularization approach might be needed in order to increase robustness. Fourth, the presence of high noise can cause issues of estimator bias. This section explores these issues, and finalizes with instructions on how to develop a Bayesian estimator.

A. Initialization

A suitable initialization for Level-set RHMs might be difficult when little information is present a priori. For instance, we may not know the starting position, what the extent is, and how detailed the model needs to be to describe the target appropriately. A simple approach would be to use the first measurements to estimate the shape using a simple model, which then can be used to initialize a more complex Level-set RHM. However, the choice of this first stage model must take into account that

1) measurements are noisy,
2) points might come from inside the object boundary,
3) the object might be moving.

We propose using a circular RHM [30], a simple but robust estimator that covers these three points. In this model, the shape parameters are represented in the form of

$$\underline{x}_k^c = \begin{bmatrix} \underline{m}_k^c \\ r_k^c \end{bmatrix}, \tag{23}$$

where $\underline{m}_k^c$ represents the center, and r_k^c the radius of the estimated circle at timestep k. This representation can be easily extended to include additional information, such as a motion model.

Let ξ be a user-defined threshold of timesteps. After this threshold, the state $\underline{\boldsymbol{x}}_{\xi}^c$ can be used to initialize the parameters $\underline{\boldsymbol{x}}^p_\xi$ of a polygonal Level-set RHM in the following way. The key idea is to approximate the circle using an n-polygon, which will sample the circle along user-defined angles θ_j, $j \in \{0, ..., n-1\}$. Then, the vertices $\underline{b}_{\xi,j}$ for the polygon parameters $\underline{x}^p_\xi$ can be calculated as

$$\begin{aligned}\underline{b}_{\xi,j} &= \underline{m}^c_\xi + r^c_\xi \cdot \begin{bmatrix}\cos(\theta_j)\\ \sin(\theta_j)\end{bmatrix}\\ &= \begin{bmatrix}1 & 0 & \cos(\theta_j)\\ 0 & 1 & \sin(\theta_j)\end{bmatrix} \cdot \begin{bmatrix}\underline{m}^c_k\\ r^c_k\end{bmatrix} .\end{aligned} \tag{24}$$

(24)

It can be seen that this operation is linear in $\underline{x}^c_\xi$. Therefore, we can find a matrix $\mathbf{A}^c$ to describe this transformation, allowing for a straightforward calculation of $\underline{\boldsymbol{x}}^p_\xi$ from $\underline{\boldsymbol{x}}^c_\xi$ in the form of

$$\underline{\boldsymbol{x}}^p_\xi = \mathbf{A}^c \cdot \underline{\boldsymbol{x}}^c_\xi \, . \tag{25}$$

However, this approach introduces the following issues. First, the new state may be too certain. Second, as all vertices are generated from the same three-dimensional random vector, it may happen that vertices may be too correlated with each other. This is unwanted, as we want the vertices of the polygon to forget about the circle and incorporate information of the new shape. In order to address this issue, we increase the uncertainty of the state by adapting (25) in the following way,

$$\underline{\boldsymbol{x}}^p_\xi = \mathbf{A}^c \cdot \underline{\boldsymbol{x}}^c_\xi + \underline{\boldsymbol{r}}_c , \tag{26}$$

with $\underline{\boldsymbol{r}}_c \sim \mathcal{N}(\underline{0}; \sigma_c^2 \cdot \mathbf{I})$ as the transformation process noise, and where σ_c^2 is a user-defined parameter.

The parameters ξ and σ_c^2 should be selected based on the specific target shape and noise level. In [15], the evaluation saw convergence at $\xi \approx 75$ using a single measurement per timestep. This value is relatively high in comparison with shape fitting tracking scenarios, and this is caused by the fact that the considered shapes are filled, and thus, most measurements stem from inside the shape and contribute little new information. The variance σ_c^2 should be chosen so that the new state is highly uncertain relative to incoming measurements. For example, a proposed value is $\sigma_c^2 \approx 10^2 \cdot \mathrm{trace}(\mathbf{C}^*_w)$, using a representative measurement noise covariance matrix.

B. Distribution of the Transformation Parameter

The distribution for the transformation parameter $\boldsymbol{s}_k$ from (15), as mentioned in Section III, must be known a priori. The particular choice varies depending on the application. A common example is when measurement sources are generated only from the edges. In these cases, as a transformation parameter of 0 is equivalent to the boundary, the only transformation is $\boldsymbol{s}_k = 0$. This is an instance of shape fitting, similar to the case explored in Section III-A. In consequence, fitting can be seen as a special case of Level-set RHMs.

An interesting case that is central to this paper is when measurements are also generated from the inside of the object. A common example is the case of a flat surface being observed by a sensor from above or below, such as Fig. 4. In general, this leads to measurement sources, represented as $\underline{z}$, being uniformly distributed in the shape. Tight group targets, where a minimum distance between participants must be held, also come close to a uniform distribution. However, of interest for the transformation parameter of Level-set RHMs, as seen in (15), is the distribution of the measurement sources after being propagated through the shape function. Thus, it makes sense to evaluate how the random variable $\boldsymbol{s}_k$, or in particular $\phi(\underline{x}^p_k, \underline{z})$, behaves when $\underline{z}$ is uniformly distributed. This then can be used as a starting point for cases when $\underline{z}$ has other distributions. The remainder of this section explores these considerations.

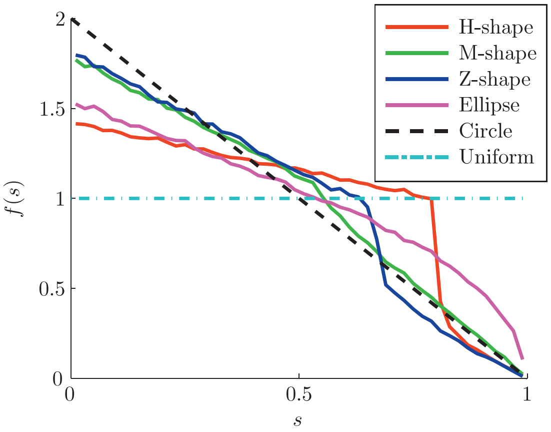


Fig. 7. Distribution of transformation parameter.

First, the distribution of $\boldsymbol{s}_k$ with a circle and the signed Euclidian distance is considered. In [15], it was shown that if measurement sources are uniformly distributed, and the scaling parameter $\boldsymbol{s}^c_k$ is 0 at the center, it follows that $(\boldsymbol{s}^c_k)^2$ is uniformly distributed as well. In this paper, the transformation parameter is 0 at the boundary, and thus, $(1 - \boldsymbol{s}_k)^2$ is uniform instead. It follows that for circles, the distribution of $\boldsymbol{s}_k$ is

$$f^s_k(s_k) = \begin{cases} 2 - 2s_k & \text{for } s_k \in [0, 1] \\ 0 & \text{otherwise.} \end{cases} \tag{27}$$

Subsequently, the distribution of other shapes was analyzed using uniformly random sampling with 10^6 points from inside the shape, and producing a histogram of the normalized Euclidian distances. Fig. 7 shows the results for an ellipse of aspect ratio 2:1, and the shapes introduced in Fig. 3. The main result is that all distributions are relatively close to that of a circle. For elongated shapes, or those made up by compositions of them, the distributions lean closer to uniform, and then sharply drop.

When tracking an object, the target shape is not known, and even if it was known, an exact calculation of the distribution of $\boldsymbol{s}_k$ is generally not efficient. However, based on Fig. 7, we propose a general approximation using a uniformly distributed random variable $\boldsymbol{u} \sim \mathcal{U}(0, 1)$

in the form of

$$s_k = 1 - \sqrt{\boldsymbol{u}}, \tag{28}$$

with the probability density from (27). In Fig. 7 this density appears as a dotted black line. For extremely elongated shapes, a uniform distribution for $\boldsymbol{s}_k$ might be preferred instead.

C. Regularization

As mentioned in Section V-A, the choice of the shape model may be an issue. In general, the shape to be tracked is not known in advance, and as such, it may be difficult to predict how detailed the representation should be. In addition, even in cases of low noise, tracking might fail because of an unsuitable initialization. These factors raise the need for a correction mechanism, and in this section, a simple form of regularization is presented.

In order to develop an approach to address these issues, we followed some ideas of active contour models [35], which face similar problems. In active contours, the idea is to minimize what is termed energy. Of particular interest for regularization is the concept of the internal energy, which is a mathematical measure of the stretch and curvature. Informally, the objective of internal energy minimization is to make the shape boundary smoother and flatter. As such, the effect of noise and overfitting is greatly reduced, and the consequences of an improper initialization can be compensated.

For polygons, internal energy minimization is implemented as follows. The neighbors of vertex $\underline{b}_{k,j}$ can be seen as connected with springs, slightly pulling $\underline{b}_{k,j}$ towards $\underline{b}_{k,j-1}$ and $\underline{b}_{k,j+1}$. In other words, $\underline{b}_{k,j}$ is being pulled in the direction of $(\underline{b}_{k,j-1} - \underline{b}_{k,j})$ and $(\underline{b}_{k,j+1} - \underline{b}_{k,j})$. This pull is measured using the regularization factor $c_k \in [0, 1]$. Thus, the pull has the form

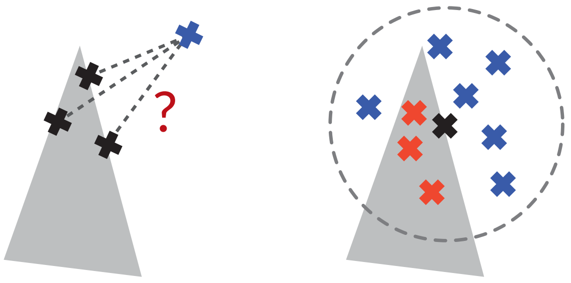
$$\begin{aligned} \underline{b}_{k+1,j} &= \underline{b}_{k,j} + c_k \cdot (\underline{b}_{k,j-1} - \underline{b}_{k,j}) + c_k \cdot (\underline{b}_{k,j+1} - \underline{b}_{k,j}) \\ &= c_k \cdot \underline{b}_{k,j-1} + (1 - 2c_k) \cdot \underline{b}_{k,j} + c_k \cdot \underline{b}_{k,j+1}. \end{aligned} \tag{29}$$

In effect, this ends up making the shape flatter, in the sense that it becomes closer to a circle. The implementation shown in (29) has the extremely useful effect that the evolution is linear in relation to $\underline{b}_{k,j}$ and its neighbors. Thus, it can be modeled using a matrix $\mathbf{A}^r(c_k)$, in the form

$$\underline{\boldsymbol{x}}_k^{reg} = \mathbf{A}^r(c_k) \cdot \underline{\boldsymbol{x}}_k, \tag{30}$$

which regularizes the parameters in $\underline{x}_k^p$ linearly and leaves the remaining terms intact. An implementation is therefore straightforward as part of the prediction step described in Section V-E.

While regularization yields a more robust estimator, this approach has some drawbacks, which should be considered when selecting a value for c_k. As regularization pushes the shape towards a more circular form, c_k should be set as low as possible (i.e., very close to zero), or else the shape details will be lost. It follows that when regularization is active, the estimator usually does not fully converge to the true shape. Nonetheless, as the pose is usually more important than an exact shape estimate, regularization is still effective even with these tradeoffs. In order to choose c_k, the levels of measurement noise, process noise, and shape curvature for the tracking scenario should be considered. For the Z-shape, we observed that values around $c_k \approx 10 \cdot \mathrm{trace}(\mathbf{R}_k)$ were good starting points.



(a) Potential sources.

(b) Measurement distances.

Fig. 8. Illustration of source of bias: source cannot be modeled correctly, and expected distances are not necessarily 0.

D. Bias in the Measurement Equation

The measurement function (6) is useful because it does not assume that the source is explicitly known. However, this simplified model also introduces an issue of bias. The origin of this problem is twofold. First, the true source is not known, as shown in Fig. 8(a). Instead, shape functions usually use a point in $\mathcal{S}(\underline{x}_k^p)$ as reference, for example in the case of the Mahalanobis distance used in (4). However, this reference point does not generally correspond to the true source. Second, even if the true source was known, (6) associates the boundary shape function to the value of 0. However, as Fig. 8(b) shows, in case of nonzero curvature or corners, the expected shape distance generally cannot be 0, as the regions with positive (red measurements) and negative distances (blue measurements) are of different sizes.

Reference [27] has proposed an approach to alleviate this problem given a measurement $\underline{y}_k$. First, a representative state $\underline{x}_k^{p,*}$ is drawn from $\underline{\boldsymbol{x}}_k^p$, such as the mean $\mathrm{E}[\underline{\boldsymbol{x}}_k^p]$. Second, one single assumed source $\underline{z}_k^*$ is taken as $S(\underline{x}_k^{p,*})$, for example as the Mahalanobis projection. For the sake of implementation, the Mahalanobis projection can be obtained in a similar fashion as (21) for $\mathbf{\Sigma} = \mathbf{C}_k^w$, by taking the point $\underline{z}_j^\pi$ with minimal distance. Third, we model the shape function values we would expect from this source, in the form of the random variable

$$\boldsymbol{v}_k := \phi^b(\underline{x}_k^{p,*}, \underline{z}_k^* + \underline{\boldsymbol{w}}_k) \,. \tag{31}$$

Finally, we associate the shape function value $\phi^b(\underline{x}_k^p, \underline{y}_k)$ to $\boldsymbol{v}_k$ in the measurement equation

$$\begin{aligned} h(\underline{x}_k, \underline{y}_k, v_k) &:= \phi^b(\underline{x}_k^p, \underline{y}_k) - v_k \\ &= \; 0 \,, \end{aligned} \tag{32}$$

where υ_k is a noise term drawn from the $\boldsymbol{\upsilon}_k$. Note that the strength of this approach depends on how well $\underline{z}_k^*$ and $\boldsymbol{\upsilon}_k$ can be approximated, and how close $\underline{x}_k^{p,*}$ is to the true shape.

Using these, we can interpret the expression in (32) as the shape function in (15), leading to the bias-alleviated RHM measurement equation

$$\begin{aligned} h(\underline{x}_k, \underline{y}_k, \upsilon_k, s_k) : &= s_k \cdot \phi_{\max}(\underline{x}_k^{p,*}) - (\phi^b(\underline{x}_k^p, \underline{y}_k) - \upsilon_k) \\ &= s_k \cdot \phi_{\max}(\underline{x}_k^{p,*}) - \phi^b(\underline{x}_k^p, \underline{y}_k) + \upsilon_k \\ &= 0. \end{aligned} \tag{33}$$

In situations with high noise, the selection of $\underline{z}_k^*$ may be relatively straightforward for convex shapes. However, for the nonconvex shapes treated in this work, an appropriate source selection and its propagation through (31) can be very challenging in case of high noise. Note that estimation issues with high noise for nonconvex shapes are not unique to this work, and appear in a wide variety of fields such as computer vision [36] and errors in variables [37, 38]. This places a limit on how well the proposed polygonal parameterization, or in practice any detailed parameterization, can work with high noise levels. In these cases, Level-set RHMs with simpler and more robust convex shapes can be used as an approximation, such as ellipses or regular polygons.

E. Prediction Step

As a summary of the section, the following subsections describe how to develop a Bayesian estimator. The prediction step is straightforward, applying the concepts described in Section V-C. Thus, (2) is written as

$$\begin{aligned} \underline{\boldsymbol{x}}_{k+1} &= a_k(\underline{\boldsymbol{x}}_k, \underline{\boldsymbol{r}}_k) \\ &= a_k^*(\mathbf{A}^r(c_k) \cdot \underline{\boldsymbol{x}}_k) + \underline{\boldsymbol{r}}_k \ , \end{aligned} \tag{34}$$

with $\underline{\boldsymbol{r}}_k \sim \mathcal{N}(\underline{0}; \mathbf{R}_k)$ as the additive Gaussian process noise, $\mathbf{A}^r(c_k)$ as the regularization matrix from (30), and $a_k^*(\cdot)$ as the system function that describes the time evolution. Unless a more detailed model is needed, $a_k^*(\cdot)$can be simply assumed to be the identity.

F. Filter Step

In order to estimate a target shape, the hidden state parameters must be inferred from the observed measurements. A common approach to achieve this is to develop a probabilistic model, i.e., a likelihood function that associates the received measurements to the state parameters.

Let $\mathcal{Y}_k = \{\underline{y}_{k,1}, \ldots, \underline{y}_{k,l}\}$ denote the set of received measurements, and $f_k^L(\underline{x}_k)$ the likelihood function, i.e., $p(\mathcal{Y}_k|\underline{x}_k)$. As it is assumed that the measurement noise and transformation parameters are independent from each other and the state, the likelihood can be rewritten as the product

$$f_k^L(\underline{x}_k) = \prod_{0 \le i \le l} f_{k,i}^L(\underline{x}_k) \tag{35}$$

of the likelihood functions $f_{k,i}^L(\underline{x}_k)$ of the individual measurements $\underline{y}_{k,i}$, i.e.,$p(\underline{y}_{k,i}|\underline{x}_k)$. This allows us to treat the likelihood of each measurement independently of each other. The likelihood (35) can be used, for example, in a Bayesian estimator to update the prior$f_k^p(\underline{x}_k)$, in the form of

$$f_k^e(\underline{x}_k) \propto f_k^L(\underline{x}_k) \cdot f_k^p(\underline{x}_k), \tag{36}$$

where $f_k^e(\underline{x}_k)$ represents the posterior.

When separated according to (35), it can be seen that incorporating all measurements simultaneously is equivalent to incorporating individual measurements sequentially in arbitrary order. Note that, in practice, this validity of this statement is determined by how well the estimator can describe the density $f_k^e(\underline{x}_k)$, which is generally merely an approximation. Thus, in most tracking scenarios with nonlinear measurements, the order in which measurements are incorporated will matter when using individual updates. Fortunately, the results will still converge to the target shape in the general case. We drop the index i once more for legibility.

As seen in (15), we do not work directly with the measurement $\underline{y}_k$, but instead with a pseudomeasurement based on $h(\cdot, \cdot, \cdot, \cdot)$. Because of this, describing the likelihood $f_k^L(\underline{x}_k)$ requires some consideration. The following subsections describe how to realize a measurement update using linear regression Kalman filters (LRKFs) and using an approximated explicit likelihood.

1) *Linear Regression Kalman Filters*: The key idea for LRKFs is to assume that the state is Gaussian distributed, i.e., $f_k^p(\underline{x}_k) = \mathcal{N}(\underline{x}_k; \underline{x}_k^p, \mathbf{C}_k^p)$, and that the state and the pseudomeasurement are jointly Gaussian. From (14), we obtain the propagated random variable

$$\boldsymbol{h}_k := h(\underline{\boldsymbol{x}}_k, \underline{y}_k, \underline{\boldsymbol{w}}_k, \boldsymbol{s}_k), \tag{37}$$

with mean $\hat{h}_k$, scalar variance $\mathbf{C}_k^h$, and cross-covariance $\mathbf{C}_k^{xh}$ with the state.

These values can be approximated using sample-based propagation approaches. An example implementation follows. Let there be a set of n samples in the form $[\, \underline{x}_{k,j}^*, \underline{w}_{k,j}^*, s_{k,j}^* \,]$ with weights $\beta_{k,j}$ for $1 \le j \le n$, drawn from $[\, \underline{\boldsymbol{x}}_k, \underline{\boldsymbol{w}}_k, \underline{\boldsymbol{s}}_k \,]$. These can be obtained, for example, by using the techniques applied in filters like the unscented Kalman filter (UKF) [39] or the smart sampling Kalman filter (S^{2}KF) [40]. Then, we define the propagated sample $h_{k,j}^*$ as

$$h_{k,j}^* := h(\underline{x}_{k,j}^*, \underline{y}_k, \underline{w}_{k,j}^*, s_{k,j}^*), \tag{38}$$

from which we obtain

$$\hat{h}_k = \sum_{j=1}^{n} \beta_{k,j} \cdot h_{k,j}^*, \tag{39}$$

$$C_k^h = \sum_{j=1}^{n} \beta_{k,j} \cdot (h_{k,j}^* - \hat{h}_k)^2 \tag{40}$$

and

$$\mathbf{C}_k^{xh} = \sum_{j=1}^{n} \beta_{k,j} \cdot (\underline{x}_{k,j}^{*} - \underline{\hat{x}}_k^{p}) \cdot (h_{k,j}^{*} - \hat{h}_k). \tag{41}$$

From these, we obtain the Kalman gain

$$\mathbf{K}_k := \mathbf{C}_k^{xh} \cdot (\mathbf{C}_k^{h})^{-1}, \tag{42}$$

leading to the updated state $f_k^e(\underline{x}_k) = \mathcal{N}(\underline{x}_k; \underline{\hat{x}}_k^e, \mathbf{C}_k^e)$ with

$$\underline{\hat{x}}_k^e = \underline{\hat{x}}_k^p + \mathbf{K}_k \cdot (\hat{h}_k - 0) \tag{43}$$

and

$$\mathbf{C}_k^e = \mathbf{C}_k^p - \mathbf{K}_k \cdot (\mathbf{C}_k^{xh})^T. \tag{44}$$

These are, in essence, the Kalman formulas, where the pseudomeasurement in (43) is 0.

It is also possible to process multiple measurements in batch. In this case, (38) can be extended to contain the propagated samples of each measurement stacked vertically as the vector $\underline{h}_{k,j}^{*}$. The remaining formulas can be extended analogously. Note that the order of measurements within a batch update will not change the result.

2) *Approximated Explicit Likelihood*: Unlike LRKFs, estimators such as particle filters [41] or the progressive Gaussian filter [42] require an explicit likelihood. Using the bias-alleviated form (33), we can see that s_k and v_k appear as additive noise terms. Thus, we define the random variable

$$\boldsymbol{\eta}_k := \boldsymbol{s}_k \cdot \phi_{max}(\underline{x}_k^{p,*}) + \boldsymbol{v}_k, \tag{45}$$

with probability density function $f_k^{\eta}(\eta_k)$. This leads to the likelihood

$$f_k^L(\underline{x}_k) = f_k^{\eta}(\phi^b(\underline{x}_k^p, \underline{y}_k)). \tag{46}$$

If $f_k^{\eta}(\eta_k)$ is untractable, a good approximation in case of low shape curvature can be obtained by assuming $\boldsymbol{\eta}_k$ is Gaussian distributed, whose mean and variance are calculated using sample-based approaches as mentioned in Section V-F.1, also described in [27]. Another approach is kernel density estimation, but this requires the selection of an appropriate bandwidth parameter [43]. Multiple measurements can be processed in batch by multiplying their likelihoods. For multiple object tracking, a mechanism for measurement-to-target association can be derived by using ideas from [44].

VI. EVALUATION

In this section, we evaluate how well polygonal Level-set RHMs can track arbitrary, nonconvex shapes in a variety of conditions. First, the capabilities of the models are evaluated using synthetic data with isotropic noise. Then, we introduce a real-life application by tracking a Z-shaped extended object with data captured by an RGBD sensor.

An important part of the experiments is to measure how the estimated shape $\mathcal{S}(\underline{x}_k^p)$ converges to the target shape S_G. For this purpose, we introduce the concept of area error, which shows how mismatched the estimated shape and the ground truth are. Let the symmetric difference between both shapes be denoted as

$$S(\underline{x}_k^p)\ \Delta\ \mathcal{S}_G := (S(\underline{x}_k^p) \cup \mathcal{S}_G) \setminus (S(\underline{x}_k^p) \cap \mathcal{S}_G), \tag{47}$$

i.e., the union of the shapes minus their intersection. In addition, let $\|\mathcal{S}(\underline{x}_k^p) \Delta\ \mathcal{S}_G\|$ be defined as the area of the symmetric difference. To normalize it, it is then divided by the area of the target shape, i.e., $||\mathcal{S}_G||$. The area error $\epsilon(k)$ is then

$$\epsilon(k) := \frac{\left\| S(\underline{x}_k^p)\ \Delta\ \mathcal{S}_G \right\|}{\|\mathcal{S}_G\|}. \tag{48}$$

A. Evaluation with Synthetic Data

This part of the evaluation consists of tracking the shapes introduced in Fig. 3 using synthetic data. Two scenarios are explored. In the first scenario, the shape of a static target is estimated. In the second scenario, the shape and position of a moving target are tracked. For backward inference, the sample-based estimator S^{2}KF [40] is used with a sampling factor of 2 times the number of dimensions. This estimator is based on the LRKF approach proposed in Section V-F.1. For the random variable S_k from (15), the approximation suggested in (28) is used.

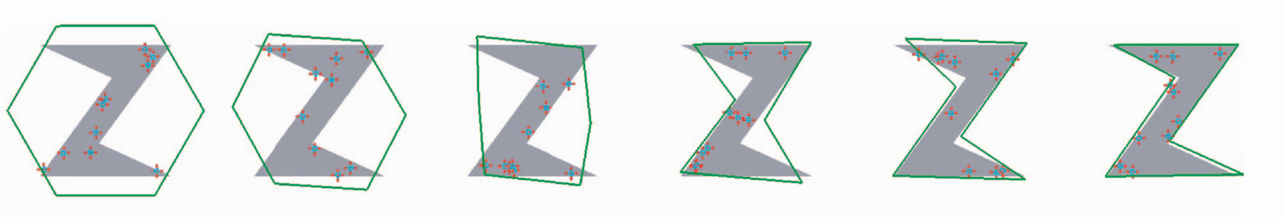
For each scenario, in each timestep k, a single measurement source is generated uniformly from the target shape $\mathcal{S}_G$. This source is then corrupted using zero-mean, Gaussian-distributed noise with covariance matrix $\mathbf{C}_k^w$. For the measurement noise $\mathbf{C}_k^w$, the three following categories of isotropic noise are used:

- low noise of $10^{-4} \cdot \mathbf{I}$,
- medium noise of $10^{-3} \cdot \mathbf{I}$,
- high noise of $10^{-2} \cdot \mathbf{I}$.

In addition, we assume a Gaussian distributed process noise with zero-mean and covariance matrix $\mathbf{R}_k$ at each timestep.

The $\underline{\boldsymbol{x}}_k$ is assumed to be Gaussian, with mean $\underline{\hat{x}}_k$ and covariance matrix $\mathbf{C}_k^x$. $\underline{\boldsymbol{x}}_0$ is initialized in the following way. For $\underline{\hat{x}}_0^p$, the vertices of the polygon are positioned so as to approximate a circle of radius 2 m, the particular configuration depending on polygon size. For clarity, a polygon of size n is referred as an n-polygon. Other state parameters, if present, are initialized with $\underline{0}$. The covariance matrix is initialized as $\mathbf{C}_0^x = 10^{-2} \cdot\ \mathbf{I}$.

1) *Static Target*: The objective is to see how well the polygonal Level-set RHM can track a static shape. For the first experiment, a Z-shape is selected as the target shape. For size reference, the shape completely fits into a square box of 2 m length. As shown in Fig. 7, the used distribution of $\boldsymbol{s}_k$ does not match the distribution of distances in the Z-shape. In order to compensate for this, a weak regularization factor of $c_k = 10^{-4}$ is used. A Gaussian distributed process noise with zero-mean and covariance matrix $\mathbf{R}_k = 10^{-5} \cdot \mathbf{I}$ is assumed at each timestep. The target is tracked using a 6-polygon. Fig. 9 shows an example representative run.



(a) $k = 0$. (b) $k = 200$. (c) $k = 400$. (d) $k = 600$. (e) $k = 800$. (f) $k = 1000$.

Fig. 9. Representative run. Timestep is k, measurements in red, uncertainty ellipse of 1- σ in cyan.

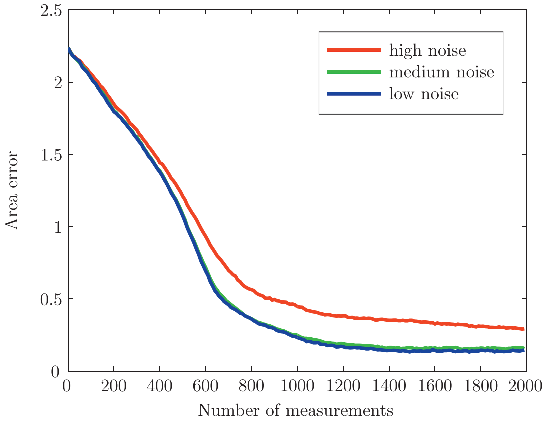
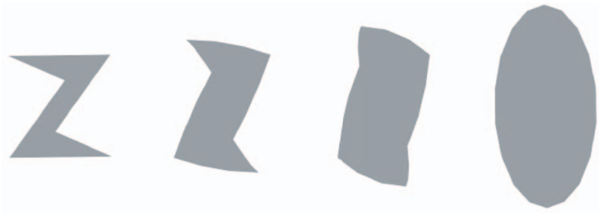


Fig. 10. Area error while tracking static Z-shape, averaged for 20 runs.

Fig. 10 shows the averaged results of the area error for 20 runs. For low (blue) and medium noise (green), the shape has converged after 1100 measurements, with an area error of around 0.09 and 0.14, respectively. For high noise (red), the error remains relatively high between 0.40 and 0.50. After almost 1200 measurements, the error remains constant, but considerably above 0. This is a consequence of the fact that, on the one hand, the distribution of $\boldsymbol{s}_k$ is an approximation, while on the other hand, the regularization procedure will always slightly pull the shape into a smoother, but incorrect, form. Even so, the results are extremely close.

One thing that stands out when looking at Fig. 10 is the high number of measurements required for convergence. This can be explained with the following three observations. First, in contrast to traditional shape fitting approaches, the target is filled, i.e., measurements also come from the interior. This means that these measurements contribute little information about the boundary. Second, the shape is nonconvex, requiring a larger state parameterization. For simpler shapes such as ellipses, Level-set RHMs are equivalent to Ellipse RHMs [31], which also converge with little measurements. Third, the starting shape is convex, so that the estimator needs considerable information in order to find the concave parts. A representative run can be seen in Fig. 9.

The experiments are then repeated using the M-shape and H-shape, using a 5-polygon and a 14-polygon, respectively. Note that the H-shape has only 12 vertices, and thus, the shape parameters are overfitted. The results after 2000 measurements are shown in Fig. 20, Fig. 21, and Fig. 22. The estimated shapes are in green, the target shapes in gray, example measurements in red, and in cyan a circle of radius corresponding to the standard deviation of the measurement noise. Fig. 20 shows the results using high noise, Fig. 21 using medium noise, and Fig. 22 using low noise.



(a) $\alpha = 0^o$. (b) $\alpha = 15^o$ (c) $\alpha = 30^o$ (d) $\alpha = 45^o$

Fig. 11. Experiment setup and results for representative run. Example measurements in red, estimated shape in green, trajectory in light blue.

2) *Moving Target*: The objective is to see how well the polygonal Level-set RHM can track a moving shape that is also morphing. The experiment setup is shown in Fig. 12. It consists of the following parts.

- The shape rotates around a circle of radius 6 m, clockwise, starting at the positive x-axis.
- Every 1 deg in the trajectory corresponds to 33 timesteps. This means that the shape moves 0.03 deg for each timestep.
- The shape is also morphing. At the start of the morph is a Z-shape, which then turns into an ellipse of height 2 m and width 1 m. Fig. 11 shows the morphing stages, which go on back and forth every 90 deg in the trajectory, or 3000 measurements. Note that the filter step only processes a single measurement at each timestep. This means that the shape is moving between measurements, and thus, every single measurement comes from the shape at a different pose.

A moderate regularization factor of $c_k = 6 \cdot 10^{-4}$ is used. In addition, a Gaussian distributed process noise with zero-mean and covariance matrix $\mathbf{R}_k = 3 \cdot 10^{-4} \cdot \mathbf{I}$ is assumed at each timestep. A 6-polygon was used to track the shape. The state parameters are extended with a constant velocity model, i.e., a new two-dimensional state

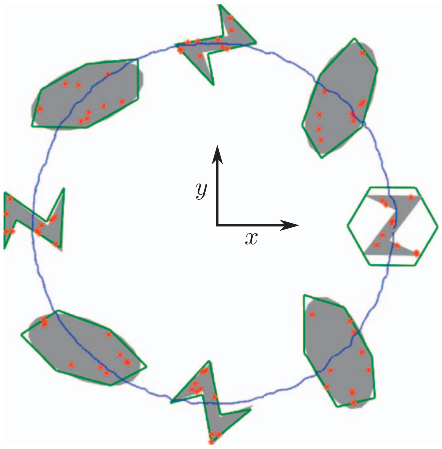


Fig. 12. Area error while tracking morphing shape, averaged for 30 runs.

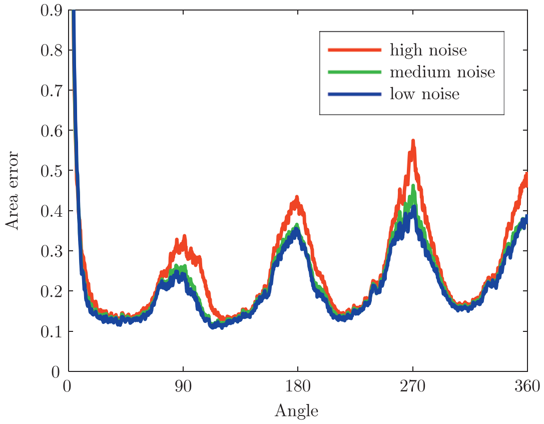


Fig. 13. Morphing stages from Z-shape to ellipse.

parameter $\underline{\boldsymbol{x}}_k^{vel}$ is appended to the state vector. This yields a dynamic model, as mentioned in (34), for each polygon vertex $\underline{b}_{k,j}$ in the shape parameters $\underline{x}_k^p$, described as

$$a_k^*(\underline{b}_{k,j}) = \underline{b}_{k,j} + \underline{x}_k^{vel} \cdot \Delta t, \tag{49}$$

where Δt is the time difference. In addition, the morphing is not considered as part of the dynamic model in order to explore the capabilities of our regularization approach from Section V-C.

In Fig. 12, the estimated shapes are in green, the ground truth in gray, example measurements in red, and the mean of the estimated polygon points in blue. In Fig. 13, the average area errors for 30 runs in function of the angle are shown. The morphing stages are evident, as each transformation into the nonconvex Z-shape leads to a peak in the area error. The ellipse, being a convex shape, can be effortlessly described (at 45^o, 135^o, 225^o, and 315^o) even with high noise, while the difficulty to properly describe the Z-shape (at 90^o, 180^o, 270^o, and 360^o) is clear again. Even then, the results are consistent with Fig. 10 and are indicative of the estimator adapting appropriately to the motion and morphing changes, even if the last one was not modeled explicitly. This can also be seen in the path of the mean of the polygon points. Even at high noise levels, at no point did it move farther than 20 cm from the correct path, and for low noise levels it constantly remained within 10 cm. For the high noise runs, however, it can be seen that the errors accumulate in successive cycles, causing the shape estimation to slowly diverge. Again, we can see that a relatively high amount of measurement is needed to estimate the shape. However, we would like to emphasize that each measurement comes from the shape at a different pose, This means that polygonal level-sets can gather a high amount of shape information throughout time even if each timestep provides few measurements and the target keeps moving.

B. Evaluation with RGBD Data

This part of the evaluation consists of tracking a real-life Z-shape extended object, using data captured by a Microsoft Kinect device. The objective is to show how well Level-set RHMs work in a real environment. First, the noise model for these devices is explained. Then, the difficulties of introduced artifacts and a gating approach are explored. Finally, in a similar fashion to Section VI-A, Level-set RHMs are evaluated using a static and a moving Z-shape object. Note that, while Kinect devices yield measurements in three dimensions, this section only uses two-dimensional data.

1) *Noise Model*: The noise model for Kinect devices is taken from previous work [12]. For the sake of completeness, the key points are explained in the following section. Kinect devices observe measurements as a depth value at a given pixel position. Let $\underline{\boldsymbol{p}}_k$ be a two-dimensional random vector describing the pixel position, and $\mathbf{d}_k^z$ be a random scalar describing the depth value. It is assumed that both variables are independent of each other.

The process of turning these values into world coordinates is called unprojection, and works as follows. Let the inverse intrinsic calibration parameters of the Kinect camera be described with the matrix $\mathbf{F}_k$ and translation vector $\underline{f}_k$, in the form of

$$\mathbf{F}_k = \begin{pmatrix} f_0 & f_1 \\ f_2 & f_3 \\ 0 & 0 \end{pmatrix}, \text{ and } \underline{f}_k = \begin{bmatrix} f_4 \\ f_5 \\ 1 \end{bmatrix}. \tag{50}$$

Then, the observed measurement in world coordinates $\underline{\boldsymbol{y}}_k^W$ are

$$\underline{\boldsymbol{y}}_k^W = (\mathbf{F}_k \cdot \underline{\boldsymbol{p}}_k + \underline{f}_k) \cdot \mathbf{d}_k^z. \tag{51}$$

Note that $\underline{\boldsymbol{y}}_{k,i}^W$ is three dimensional. Finally, as polygonal RHMs require the measurement to be two dimensional, the observed measurement $\underline{\boldsymbol{y}}_k^W$ is projected onto a plane using a 2 × 3 projection matrix $\mathbf{P}_k$. Thus, the final form of measurement $\underline{\boldsymbol{y}}_k$ is

$$\underline{\boldsymbol{y}}_k = \mathbf{P}_k \cdot \underline{\boldsymbol{y}}_k^W, \tag{52}$$

with corresponding mean $\underline{\hat{y}}_k$ and covariance matrix $\mathbf{C}_k^y$. From these, we derive the parameters for the measurement

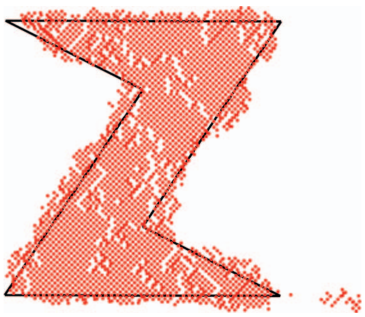


Fig. 14. Moving Z-shape as observed by a Kinect. Target shape boundary in black, measurements in red. Note presence of outliers.

noise $\underline{\boldsymbol{w}}_k \sim \mathcal{N}(\underline{0}, \mathbf{C}_k^y)$. Note that the noise term $\underline{\boldsymbol{w}}_k$ is, in general, not isotropic.

2) *Gating*: The main issue separating the Kinect evaluation from the synthetic data evaluation is that observing the tracked object introduces artifacts. In consequence, on the one hand, some parts of the object are not observed correctly, while on the other hand, spurious measurements (outliers) are observed that do not correspond to any source. This makes it very difficult to estimate the distribution of measurement sources. In practice, given the closed nature of the Kinect internals, it is impossible to correctly model these effects. Fig. 14 visualizes this challenge, with measurements in red and the ground truth in black.

This issue is particularly troubling when the tracked object is moving, because of the high amount of outliers. This raises the need for a gating mechanism. A simple gating algorithm for Level-set RHMs is introduced as follows.

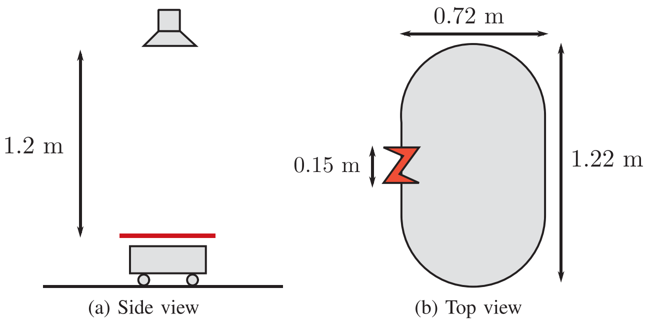
At timestep k, a measurement $\underline{y}_k$ arrives, modeled as having been disturbed with additive noise $\underline{\boldsymbol{w}}_k$. The measurement and noise are then propagated through the measurement function (15), or its bias-alleviated form (33). As described in Section V-F.1, let $\hat{h}_k$ be the mean and $\mathbf{C}_k^h$ the scalar variance of this propagated distribution. Finally, based on the one-dimensional Mahalanobis distance, we accept the measurement $\underline{y}_k$ only if

$$(\mathbf{C}_k^h)^{-1} \cdot \left(\hat{h}_k - 0\right)^2 < d_G^2, \tag{53}$$

where 0 represents the pseudomeasurement, and d_G is a user-defined distance threshold which indicates how close the measurement has to be to the estimated shape.

A more intuitive approach is to establish which fraction of potential measurements we want to incorporate. Thus, let d_G^ϵ in [0, 1] be a user-defined gating threshold. For example, $d_G^\epsilon = 0.99$ indicates that we want to accept 99% of potential measurements. Then, as $\boldsymbol{h}_k$ is assumed as Gaussian distributed, the left side of (53) can be assumed as chi-squared distributed. This allows us to determine d_G^2 by using

$$d_G^2 = \text{chi2inv}(d_G^\epsilon,\ 1), \tag{54}$$



(a) Side view (b) Top view

Fig. 15. Experiment setup for Kinect evaluation.

where chi2inv($\cdot$, 1) represents the inverse of the cumulative chi-squared distribution with 1 degree of freedom.

3) *Experiment Setup*: The experiment setup (Fig. 15) consists of a toy train moving on a track on the *XY* plane. On top of the train, a Z-shaped extended object is attached, with width and height of 0.15 m. The camera observes the object at around a distance of 1.2 m. The objective is to track this Z-shaped object as it moves.

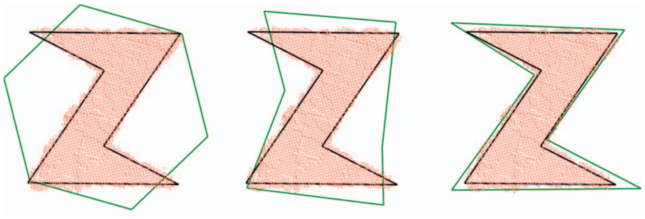
For the sake of reproducibility, the experiment parameters are as follows. The inverse camera intrinsic parameters are

$$\mathbf{F}_k = 1.880 \cdot 10^{-3} \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}, \text{ and } \underline{f}_k = \begin{bmatrix} -0.601 \\ -0.451 \\ 1 \end{bmatrix},$$

using a depth image of size 640 × 480 pixels. For the measurement, both $\underline{\boldsymbol{p}}_k$ and $\mathbf{d}_k^z$ are assumed to be Gaussian distributed. The corresponding means $\hat{\underline{p}}_k$ and $\hat{d}_k^z$ are directly taken from the values measured by the Kinect. For the pixel position, a covariance matrix of $\mathbf{C}_k^p = \frac{1}{3} \cdot \mathbf{I}$ pixels^2 is assumed. The depth scalar variance $\mathbf{C}_k^{d^z}$ depends on the observed depth [12]. However, in the neighborhood of 1.2 m, the value for this variance is about 10^{-5} m^2. Finally, the matrix $\mathbf{P}_k$ is simply a projection matrix to the plane *XY*. For a reference on the uncertainty of $\underline{\boldsymbol{w}}_k$, both eigenvalues of $\mathbf{C}_k^y$ are always in the order of magnitude of 10^{-6} m^2.

4) *Static Tracking*: In this section we evaluate how well a polygonal Level-set RHM can track a real-life static shape, given the conditions described in Section VI-B.2. For the experiment, the camera observes the static Z-shaped object from above (15a). The first frame is used for initialization by means of a circle estimator (Section V-A), with a one-time Gaussian transformation process noise with $\mathbf{R}_\xi = 10^{-4} \cdot \mathbf{I}$ m^2, and a threshold of $\xi = 50$ timesteps. Then, independent evaluation runs are launched, where each run is initialized with the aforementioned circle, and uses a different frame.

The measurements to be used are taken at random from this frame, in order to ensure our requirement that s_k is drawn from $\boldsymbol{s}_k$ independently between measurements. Measurements are then processed sequentially using a single measurement per timestep. Note that the estimator can also process multiple measurements per timestep if



(a) $k = 0$. (b) $k = 600$. (c) $k = 1200$.

Fig. 16. Example run. Target shape boundary in black, example measurements in red, estimate in green. Note presence of outliers.

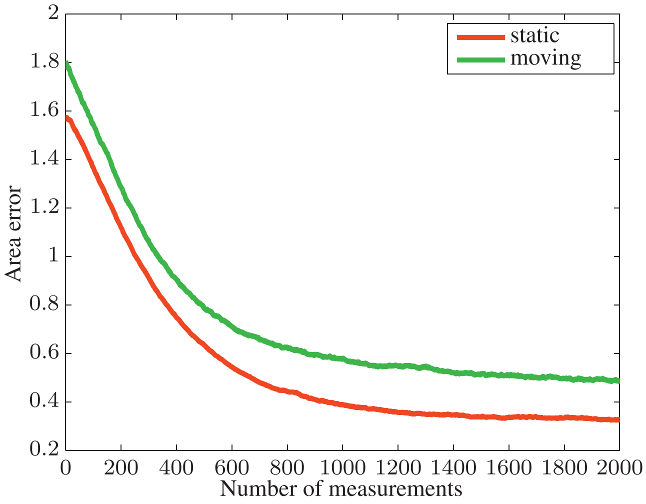


Fig. 17. Area error for Z-shape Kinect tracking.

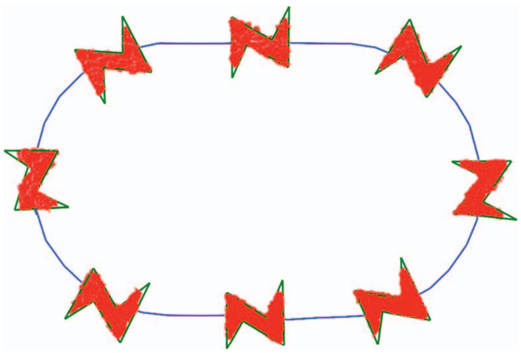


Fig. 18. Results of shape tracking at 8 snapshots. Measurements in red, estimated shape in green, trajectory in blue.

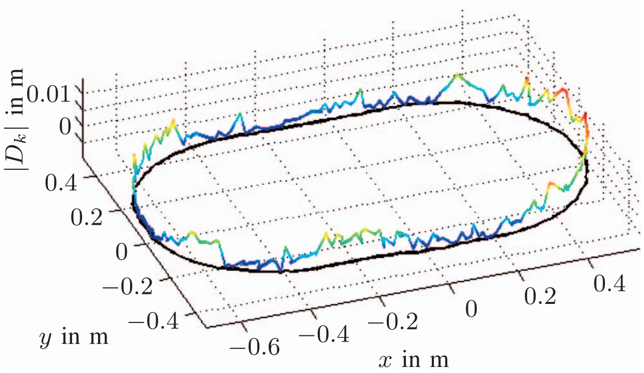


Fig. 19. Absolute deviation $|D_k|$ to ground truth path.

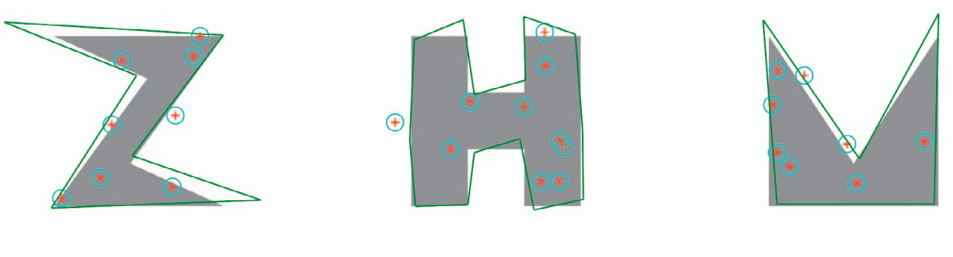
desired. A regularization factor of $c_k = 6 \cdot 10^{-5}$ is used. In addition, a Gaussian distributed process noise with zero-mean and covariance matrix $\mathbf{R}_k = 10^{-7} \cdot \mathbf{I}$ is assumed at each timestep. As the shape is static, the number of outliers is small, and therefore the gating approach of Section VI-B.2 is not used. Fig. 16 shows an example run, and Fig. 17 shows the average area error for 100 runs (red).

As can be seen in Fig. 14, there are several measurements outside of the shape that cannot be described using the noise model. Because of this, a convergence to the area errors seen in Fig. 10 is not possible. The area error for this shape is about 0.35, as seen in Fig. 17. This is still an extremely close estimate, as shown in Fig. 16(c), where it can be observed that the estimated value is simply a slightly scaled version of the correct Z-shape. This scaling can easily be explained when considering the reference picture in Fig. 14. It can also be seen that the estimator has mostly converged at about 1000 measurements. The Z-shaped object, in the depth image, was observed in a window of about 60 × 60 pixels^2, yielding about 2200 usable measurements. This indicates that, under the given circumstances, a single frame is more than enough to allow the estimator to converge.

5) *Dynamic Tracking*: The final experiment consists of evaluating how well a polygonal Level-set RHM can track a moving shape being observed by a Kinect. For this experiment, the toy train moves along the track at a speed of about $40 \frac{\text{cm}}{\text{s}}$. For reference, a Kinect observes the train in discrete intervals at a rate of 30 frames per second. The parameters for the evaluation are the same as in Section VI-B.4, except for the addition of a constant velocity model. In addition, because of the presence of outliers caused by the moving object, the gating approach from Section V–B.2 is used with $d_G^2 = 9$.

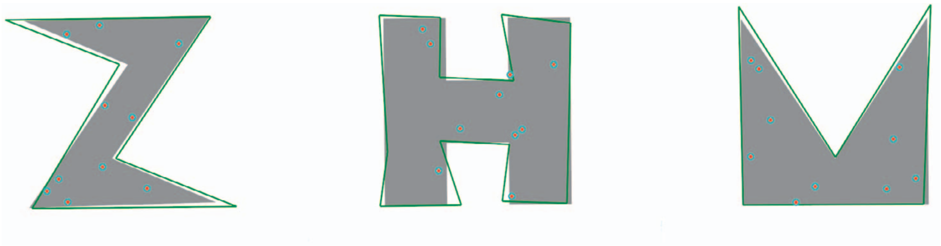
First, we evaluate how much the measurement quality has degraded when the object is moving. For this, 40 control frames were processed as described in Section VI-B.4, i.e., the Z-shape was tracked from scratch starting with a circle and then the area error to the ground truth was measured. The results are shown in Fig. 17, in green. It can be seen that the area error has increased to about 0.50. For reference, the mismatched parts in the estimate are about 40% larger than those in Fig. 16(c), which shows that the result of the estimation is still very close.

Finally, the moving object is tracked by initializing the estimator using the first frame, and then updating it using the following ones, using a total of 255 frames. Fig. 18 shows eight snapshots of the tracking experiment. Even if the area error is higher, the estimator still correctly identifies the target shape and its pose, though it is a slightly scaled form. It can also be seen that the constant velocity model has little trouble following the object, even considering the increased presence of outliers, and the fact that some corners cannot be correctly observed.



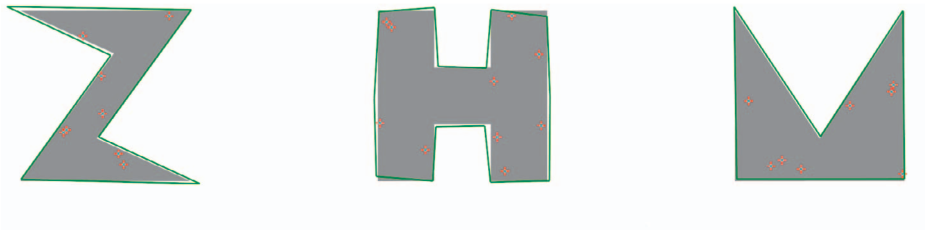
(a) Z-shape, $\epsilon = 0.40$. (b) H-shape, $\epsilon = 0.18$. (c) M-shape, $\epsilon = 0.23$.

Fig. 20. Example run for high measurement noise at $k = 2000$. Target in gray, estimate in green, example measurements in red, uncertainty of 1-σ in cyan.



(a) Z-shape, $\epsilon = 0.15$. (b) H-shape, $\epsilon = 0.095$. (c) M-shape, $\epsilon = 0.090$.

Fig. 21. Example run for medium measurement noise at $k = 2000$. Target in gray, estimate in green, example measurements in red, uncertainty of 1-σ in cyan.



(a) Z-shape, $\epsilon = 0.092$. (b) H-shape, $\epsilon = 0.069$. (c) M-shape, $\epsilon = 0.057$.

Fig. 22. Example run for low measurement noise at $k = 2000$. Target in gray, estimate in green, example measurements in red, uncertainty of 1-σ in cyan.

Fig. 19 shows the path deviation of the estimator, i.e., how much the center of the estimated shape deviated from the ground truth path. The maximum deviation was 14.9 mm, meaning that the deviation error was always under 10% of the width of the object. The deviation mean was 2.3 mm, while the mean of the absolute values of deviations was 4.8 mm.

VII. CONCLUSION AND FUTURE WORK

In this work, a Bayesian estimation algorithm for tracking arbitrary shapes was introduced in the form of Level-set RHMs. This allows the modeling and tracking of a variety of shapes, including filled shapes that are neither convex nor star-convex.

The objective was to extend the concept of RHMs in order to track arbitrary, nonconvex extended objects. The key idea was to implement shape transformation using level-sets of a given shape function. This, combined with a random variable s_k describing the probability for each level-set to generate a measurement, allowed for a simple measurement equation that is both easy to understand and to implement.

The implementation consists of describing the tracked extended object using polygons. This representation has the advantage of being efficient, simple, and widely used in research literature. The challenges of tracking arbitrary extended objects were also explored, together with approaches and compromises in order to solve them. These include an approximation for the distribution of measurement sources, an initialization procedure, and how to increase robustness by using a regularization approach.

The evaluation showed the viability of Level-set RHMs both using synthetic data, and data captured from a real-life object using a Kinect device. The results of the experiments with synthetic data were twofold. On the one hand, they displayed the limitations of the approaches and compromises we presented. On the other hand, they showed that Level-set RHMs, even with uncertain information, produce results extremely close to the expected ground truths. In particular, we demonstrated how our approach can gather shape information throughout time, even if each timestep provides very few measurements and the shape is constantly moving. This robustness was also shown in the evaluations using Kinect data, which showed that Level-set RHMs can be used in real-life scenarios with noisy sensors, and produce close estimates even in the presence of artifacts and outliers.

Future work includes a further exploration of the regularization approach, e.g., ways to allow the shape to recover from an arbitrary initialization, or particularly damaging outlier measurements. In addition, work on representations with splines can also be of advantage. Another important topic is the automatic adjustment of the shape complexity, for cases of inappropriate initialization, high measurement noise, or high kinematic noise.

REFERENCES

[1] Bar-Shalom, Y., Willett, P. K., and Tian, X.
Tracking and Data Fusion: A Handbook of Algorithms. Storrs, CT: YBS Publishing, 2011.

[2] Porrill, J.
Fitting ellipses and predicting confidence envelopes using a bias corrected Kalman filter.
Image Vision Computing, **8** (1990), 37–41.

[3] Zhang, Z.
Parameter estimation techniques: A tutorial with application to conic fitting.
Image and Vision Computing, **15**, 1 (1997), 59–76.

[4] Frosio, I., and Borghese, N. A.
Real-time accurate circle fitting with occlusions.
Pattern Recognition, **41**, 3 (2008), 1041–1055.

[5] Gilholm, K., and Salmond, D.
Spatial distribution model for tracking extended objects.
IEE Proceedings on Radar, Sonar and Navigation, **152**, 5 (Oct. 2005), 364–371.

[6] Werman, M., and Keren, D.
A Bayesian method for fitting parametric and nonparametric models to noisy data.
IEEE Transactions on Pattern Analysis and Machine Intelligence, **23**, 5 (2001), 528–534.

[7] Baum, M., and Hanebeck, U. D.
Tracking an extended object modeled as an axis-aligned rectangle.
In *4th German Workshop on Sensor Data Fusion: Trends, Solutions, Applications (SDF 2009), 39th Annual Conference of the Gesellschaft für Informatik e.V. (GI)*, Lübeck, Germany, Oct. 2009.

[8] Granstrom, K.
Tracking rectangular and elliptical extended targets using laser measurements.
In *Proceedings of the 15th International Conference on Information Fusion (FUSION)*, Edinburgh, United Kingdom, 2011. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber =5977698

[9] Sun, L., Li, X., and Lan, J.
Extended object tracking based on support functions and extended Gaussian images.
In *2013 16th International Conference on Information Fusion (FUSION)*, July 2013, 1526–1533.

[10] Petrov, N., Mihaylova, L., Gning, A., and Angelova, D.
A novel sequential Monte Carlo approach for extended object tracking based on border parameterisation.
In Proceedings of the 15th International Conference on Information Fusion (FUSION), Edinburgh, United Kingdom, 2011.

[11] Chernov, N.
Circular and Linear Regression: Fitting Circles and Lines by Least Squares. Boca Raton, FL: CRC Press, 2010.

[12] Faion, F., Friedberger, S., Zea, A., and Hanebeck, U. D.
Intelligent sensor-scheduling for multi-Kinect-tracking.
In *Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2012)*, Vilamoura, Algarve, Portugal, Oct. 2012, 3993–3999.

[13] Baum, M., and Hanebeck, U. D.
Extended object tracking with random hypersurface models.
IEEE Transactions on Aerospace and Electronic Systems, to be published.

[14] Faion, F., Baum, M., and Hanebeck, U. D.
Tracking 3D shapes in noisy point clouds with random hypersurface models.
In *Proceedings of the 15th International Conference on Information Fusion (Fusion 2012)*, Singapore, July 2012.

[15] Baum, M., Noack, B., and Hanebeck, U. D.
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, July 2010.

[16] Baum, M., and Hanebeck, U. D.
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, IL, July 2011.

[17] Gilholm, K., Godsill, S., Maskell, S., and Salmond, D.
Poisson models for extended target and group tracking.
In *SPIE: Signal and Data Processing of Small Targets*, 2005.

[18] Petrov, N., Mihaylova, L., Gning, A., and Angelova, D.
Group object tracking with a sequential Monte Carlo method based on a parameterised likelihood function.
Monte Carlo Methods and Applications, 2012.

[19] Petrov, N., Gning, A., Mihaylova, L., and Angelova, D.
Box particle filtering for extended object tracking.
In *15th International Conference on Information Fusion (FUSION)*, 2012, 82–89.

[20] Feldmann, M., and Fränken, D.
Tracking of extended objects and group targets using random matrices – A new approach.
In *Proceedings of the 11th International Conference on Information Fusion (Fusion 2008)*, July 2008, 1–8.

[21] Feldmann, M., and Fränken, D.
Advances on tracking of extended objects and group targets using random matrices.
In *Proceedings of the 12th International Conference on Information Fusion (Fusion 2009)*, Seattle, WA, July 2009.

[22] Orguner, U.
A variational measurement update for extended target tracking with random matrices.
IEEE Transactions on Signal Processing, **60**, 7 (July 2012), 3827–3834.

[23] Granström, K., and Orguner, U.
A PHD filter for tracking multiple extended targets using random matrices.
IEEE Transactions on Signal Processing, **60**, 11 (Nov. 2012), 5657–5671.

[24] Lan, J., and Rong Li, X.
Tracking of extended object or target group using random matrix - Part I: New model and approach.
In *15th International Conference on Information Fusion (FUSION)*, July 2012, 2177–2184.

[25] Lan, J., and Rong Li, X.
Tracking of extended object or target group using random matrix - Part II: Irregular object.

In *15th International Conference on Information Fusion (FUSION)*, July 2012, 2185–2192.

[26] Osher, S., and Fedkiw, R.
Level Set Methods and Dynamic Implicit Surfaces, Vol. 153. New York: Springer, 2002.

[27] Faion, F., Zea, A., Baum, M., and Hanebeck, U. D.
Partial likelihood for unbiased extended object tracking.
In *Proceedings of the 18th International Conference on Information Fusion (Fusion 2015)*, Washington, DC, July 2015.

[28] Zea, A., Faion, F., and Hanebeck, U. D.
Tracking extended objects using extrusion random hypersurface models.
In *Proceedings of the IEEE ISIF Workshop on Sensor Data Fusion: Trends, Solutions, Applications (SDF 2014)*, Bonn, Germany, Oct. 2014.

[29] Faion, F., Zea, A., Baum, M., and Hanebeck, U. D.
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.

[30] Baum, M., Klumpp, V., and Hanebeck, U. D.
A novel Bayesian method for fitting a circle to noisy points.
In *Proceedings of the 13th International Conference on Information Fusion (Fusion 2010)*, Edinburgh, United Kingdom, July 2010.

[31] Baum, M., and Hanebeck, U. D.
Fitting conics to noisy data using stochastic linearization.
In *Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2011)*, San Francisco, CA, Sept. 2011.

[32] Osher, S., and Paragios, N.
Geometric Level Set Methods in Imaging, Vision, and Graphics. New York: Springer, 2003.

[33] Malladi, R., Sethian, J., and Vemuri, B.
Shape modeling with front propagation: A level set approach.
IEEE Transactions on Pattern Analysis and Machine Intelligence, **17**, 2 (Feb. 1995), 158–175.

[34] Haines, E.
Point in polygon strategies.
Graphics Gems IV, **994** (1994), 24–26.

[35] Kass, M., Witkin, A., and Terzopoulos, D.
Snakes: Active contour models.
International Journal of Computer Vision, **1** (1988), 321–331. [Online]. Available: http://dx.doi.org/10.1007/BF00133570

[36] Okatani, T., and Deguchi, K.
On bias correction for geometric parameter estimation in computer vision.
In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, 959–966. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs/_all.jsp?arnumber=5206722

[37] Kanatani, K., and Al-Sharadqah, A.
Renormalization returns: Hyper-renormalization and its applications.
In *Computer Vision-ECCV*, 2012, 384–397. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-642-33712-3\-28

[38] Griliches, Z., and Ringstad, V.
Error-in-the-variables bias in nonlinear contexts.
Econometrica: Journal of the Econometric Society, **38**, 2 (1970), 368–370. [Online]. Available: http://www.jstor.org/stable/1913020

[39] Julier, S. J., and Uhlmann, J. K.
A new extension of the Kalman filter to nonlinear systems.
In *Prooceedings of SPIE, Signal Processing, Sensor Fusion, and Target Recognition VI* (July 28, 1997), 182–193.

[40] Steinbring, J., and Hanebeck, U. D.
S2KF: The smart sampling Kalman filter.
In *Proceedings of the 16th International Conference on Information Fusion (Fusion 2013)*, Istanbul, Turkey, July 2013.

[41] Doucet, A., and Johansen, A. M.
A tutorial on particle filtering and smoothing: Fifteen years later,
Handbook of nonlinear filtering, **12** (2009), 656–704.

[42] Steinbring, J., and Hanebeck, U. D.
Progressive Gaussian filtering using explicit likelihoods.
In *Proceedings of the 17th International Conference on Information Fusion (Fusion 2014)*, Salamanca, Spain, July 2014.

[43] Sheather, S. J., and Jones, M. C.
A reliable data-based bandwidth selection method for kernel density estimation.
Journal of the Royal Statistical Society, Series B (Methodological) (1991), 683–690.

[44] Baum, M., Noack, B., and Hanebeck, U. D.
Random hypersurface mixture models for tracking multiple extended objects.
In *Proceedings of the 50th IEEE Conference on Decision and Control (CDC 2011)*, Orlando, FL, Dec. 2011.

Antonio Zea received his Dipl.-Inform. in computer science from the the Karlsruhe Institute of Technology (KIT), Germany, in 2013.

Currently, he is working towards a Ph.D. degree at the Intelligent Sensor-Actuator-Systems Laboratory, Karlsruhe Institute of Technology (KIT), Germany. His research interests are in the field of extended object tracking, shape estimation, and telepresence.

Florian Faion received his Dipl.-Inform. in computer science from the the Karlsruhe Institute of Technology (KIT), Germany, in 2010.

Currently, he is working towards a Ph.D. degree at the Intelligent Sensor-Actuator-Systems Laboratory, Karlsruhe Institute of Technology (KIT), Germany. His research interests are in the field of extended object tracking, shape estimation and telepresence.

Marcus Baum received the Diploma degree in computer science from the University of Karlsruhe (TH), Germany, in 2007, and graduated as Dr.-Ing. (Doctor of Engineering) at the Karlsruhe Institute of Technology (KIT), Germany, in 2013.

From 2013 to 2014, he was postdoc and assistant research professor at the University of Connecticut, USA. He is Assistant Professor at the University of Göttingen, Germany. His research interests are in the field of data fusion, estimation, and tracking.

Uwe D. Hanebeck obtained his Ph.D. degree in 1997 and his habilitation degree in 2003, both in electrical engineering from the Technical University in Munich, Germany.

He is a chaired professor of Computer Science at the Karlsruhe Institute of Technology (KIT) in Germany and director of the Intelligent Sensor-Actuator-Systems Laboratory (ISAS). From 2005 to 2015, he was the chairman of the Research Training Group RTG 1194 “Self-Organizing Sensor-Actuator-Networks” financed by the German Research Foundation. His research interests are in the areas of information fusion, nonlinear state estimation, stochastic modeling, system identification, and control with a strong emphasis on theory-driven approaches based on stochastic system theory and uncertainty models. Research results are applied to various application topics like localization, human-robot-interaction, assistive systems, sensor-actuator-networks, medical engineering, distributed measuring system, and extended range telepresence. Research is pursued in many academic projects and in a variety of cooperations with industrial partners.

Prof. Hanebeck is author and coauthor of more than 380 publications in various high-ranking journals and conferences.