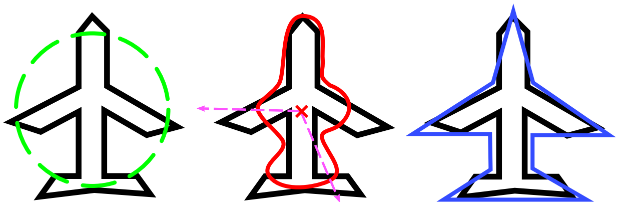
Level-Set Random Hypersurface Models for Tracking Non-Convex Extended Objects

Antonio Zea, **Florian Faion**, **Marcus Baum**, and **Uwe D. Hanebeck**
Intelligent Sensor-Actuator-Systems Laboratory (ISAS)
Institute for Anthropomatics
Karlsruhe Institute of Technology (KIT), Germany
antonio.zea@student.kit.edu, florian.faion@kit.edu, marcus.baum@kit.edu, uwe.hanebeck@ieee.org

***Abstract*—This paper presents a novel approach to track a non-convex 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 high measurement noise. In this paper, shapes are described using polygons and shape regularization is applied using ideas from active contour models.**

***Index Terms*—Random Hypersurface Models, Active Contours, Level-sets, Target tracking, Shape Tracking**



(a) Circle RHM. (b) Star-convex RHM. (c) Level-Set RHM.

Figure 1: Different RHMs.

I. Introduction

In target tracking scenarios [1], the usual conditions are high noise and only a few measurements per time step. 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, sticks, or rectangles. 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 [2]. In addition, shape tracking algorithms usually do not consider measurements corrupted by noise. Altogether, an explicit model is needed that captures the wide range of measurement quality.

Work has been done to model such scenarios in the form of Random Hypersurface Models (RHMs) [3]. Representations include approximations using circular and elliptical shapes, best suited for high noise scenarios. More detail can be captured using star-convex shapes, as shown in Fig. 1. This paper builds upon the theoretic concepts of these works and presents a generalization for arbitrary, simply connected shapes. The main contribution is a generalized RHM that uses level-sets for modeling targets.

A. Related Work

Besides the RHM approach to extended object tracking [3]–[7], there are essentially two lines of work in literature: First, particle filter methods [8]–[11] have been proposed for dealing with the nonlinearity of the estimation problem. Second, random matrix theory [12]–[18] allows to derive analytic expressions for targets that are modeled as ellipses.

B. Overview

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 Sec. II. In Sec. 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 Sec. IV. In Sec. V, the proposed approach is demonstrated and evaluated using synthetic data. Finally, this work concludes with a short summary and an outlook to future work in Sec. VI.

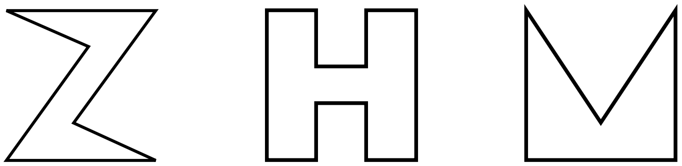
II. Modeling Extended Objects

In this section, a probabilistic model for an extended object to be tracked will be 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 system.

A. Shape model

This work focuses on tracking compact, simply connected shapes. 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 a time step 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 shape parameters are represented with



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

Figure 2: A selection of shapes.

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 $\underline{\boldsymbol{x}}_k$ will be assumed to contain $\underline{\boldsymbol{x}}_k^p$, and additional parameters as required by further models, e.g., velocity or acceleration.

B. Measurement model

At each time step k, a set of measurements $\mathcal{Y}_k = \{\underline{y}_{k,1}, \ldots, \underline{y}_{k,l}\}$ becomes available. It is assumed that all measurements $\underline{y}_{k,i}$, where $1 \leq i \leq l$, are mutually independent. Only a single measurement per time step is necessary to update the shape, and if there are multiple measurements available, they can be processed sequentially.

It is assumed that a given measurement $\underline{y}_{k,i}$ was generated by a point $\underline{z}_{k,i} \in \mathcal{S}(\underline{x}_k^p)$. The point $\underline{z}_{k,i}$ is called the *measurement source*. Associating each $\underline{y}_{k,i}$ to its generating source $\underline{z}_{k,i}$ is not straightforward, as can be seen in cases of measurement noise or where several sources are equally likely. However, we do not need to associate each measurement to its source explicitly, as the target extent will be modeled using *Random Hypersurface Models*, introduced in Sec. III.

The measurement $\underline{y}_{k,i}$ and its source $\underline{z}_{k,i}$ are generally not equal, as observation by a sensor usually introduces noise. In this paper, the noise will be assumed to be additive, zero-mean and Gaussian distributed. As such, the relation between $\underline{y}_{k,i}$ and $\underline{z}_{k,i}$ can be described as

$$\underline{y}_{k,i} = \underline{z}_{k,i} + \underline{w}_{k,i} \ , \tag{1}$$

where $\underline{w}_{k,i}$ is a realization of the random vector $\underline{\boldsymbol{w}}_{k,i} \sim \mathcal{N}(\underline{w}_{k,i}; \underline{0}, \mathbf{C}_{w_{k,i}})$.

C. Dynamic model

The dynamic model describes how the state $\underline{\boldsymbol{x}}_k$ evolves in time between successive time steps 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 a_k describes this process, and has the form

$$\underline{\boldsymbol{x}}_{k+1} = \underline{a}_k(\underline{\boldsymbol{x}}_k, \underline{\hat{u}}_k, \underline{\boldsymbol{v}}_k) \ , \tag{2}$$

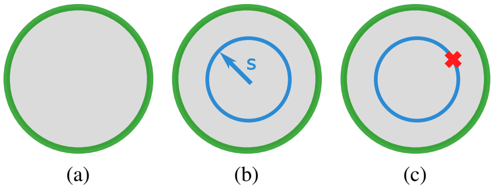
where $\underline{\hat{u}}_k$ is a known input, and $\underline{\boldsymbol{v}}_k$ the process noise. However, this paper imposes no restriction on the dynamic model.

III. Level-set RHMs

A *Random Hypersurface Model* (RHM) [3] is a probabilistic model to describe how to generate single measurement sources from an extended shape. An RHM allows to derive a functional relationship, i.e., a measurement equation, between the measurements and the shape parameters. This measurement equation paves the way to use standard nonlinear estimation techniques such as Gaussian state estimators [19] for performing a measurement update.

The key idea is to interpret a target shape by transforming an underlying shape in an arbitrary way, with a random variable $\boldsymbol{s}$. This transformation is referred to as *scaling*, and is not necessarily linear. The process of generating a measurement source is as follows. First, a *scaled* version of the underlying shape is produced by randomly choosing a scaling factor from $\boldsymbol{s}$. Then, the measurement source is selected from this scaled shape according to an arbitrary, unknown rule. The particular distribution of $\boldsymbol{s}$ has to be known in advance and can be assumed to be independent of the target shape.

The scaling factor can be interpreted in arbitrary ways, depending on the model [4] [6] [20]. An example with a circular disk follows.



(a) (b) (c)

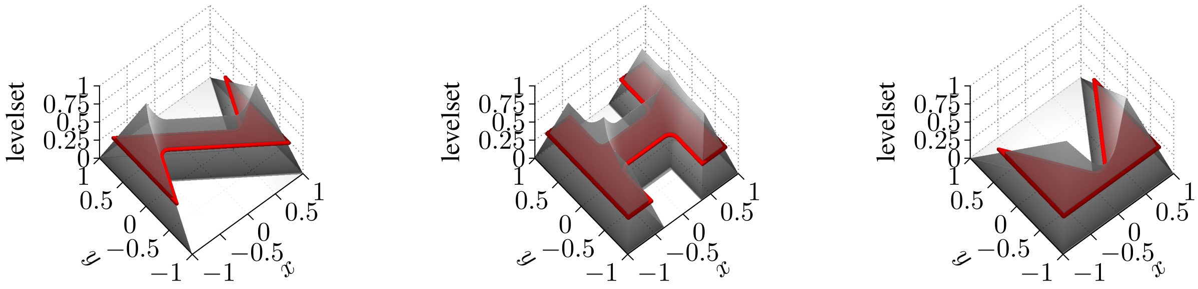
Figure 4: Random Hypersurface Model for a disk.

Example 1 (RHM with a circular disk)
Let $\mathcal{S}$ be a disk with radius r_k (Fig. 4a) centered at the origin. For a given measurement $\underline{y}_{k,i}$, we do not know its source $\underline{z}_{k,i}$ from the disk. Thus, we model the measurement as being the result of the following process. First, an unknown factor $s_{k,i}$ scales the disk boundary, as seen in Fig. 4b. The shape scaling produces a circle of radius $s_{k,i} \cdot r_k$. Then, the measurement source $\underline{z}_{k,i}$ is selected from this scaled circle (Fig. 4c), and observed by a sensor as $\underline{y}_{k,i}$. It can be seen that

$$|\underline{z}_{k,i}| = r_k \cdot s_{k,i} \ . \tag{3}$$

However, the exact $s_{k,i}$ for $\underline{y}_{k,i}$ is not known. Because of that, we model it using the random variable $\boldsymbol{s}_{k,i}$. Finally, (1) and (3) yield the following measurement function h for the measurement $\underline{y}_{k,i}$,

$$\begin{aligned} 0 &= |\underline{z}_{k,i}| - r_k \cdot s_{k,i} \\ &= |\underline{y}_{k,i} - \underline{w}_{k,i}| - r_k \cdot s_{k,i} \\ &:= h(\underline{\boldsymbol{x}}_k, \underline{y}_{k,i}, \underline{\boldsymbol{w}}_{k,i}, \boldsymbol{s}_{k,i}) \ . \end{aligned} \tag{4}$$



(a) Z-shape

(b) H-shape

(c) M-shape

Figure 3: Level-sets for the different shapes in Fig. 2.

While a scaling mechanism for circles is simple, for more complex shapes a way to implement scaling is not straightforward. Fig. 2 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 like the M-shape are clearly not convex, however, they are *star-convex*. The defining property of a star-convex shape is that there exists 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 $\mathcal{S}(\underline{x}_k^p)$. A method to scale star-convex shapes was discussed in [6]. This does not apply to shapes like the H-shape or the Z-shape, as no such point exists for them. The task is, then, to find a suitable and general scaling mechanism for these shapes.

For the Level-Set RHMs that are introduced in this paper, shape scaling is implemented using the concepts of *level-sets* of a given *shape function*. The key idea of a shape function is that, for given shape parameters $\underline{x}_k^p$, it associates every point $\underline{z}$ to a scalar in $\mathbb{R}$. A common example of shape functions are signed distance functions [21]. The choice of a shape function is arbitrary, and the only requirement imposed in this paper is for the function to be continuous. A shape function $\phi(\underline{x}_k^p, \underline{z})$ can be used to describe a shape, as can be seen in Definition 1.

Definition 1 (Implicit Representation of a Shape)
The shape $\mathcal{S}(\underline{x}_k^p)$ associated with the shape function $\phi(\underline{x}_k^p, \underline{z})$ is defined as

$$\mathcal{S}(\underline{x}_k^p) := \{\underline{z} \mid \phi(\underline{x}_k^p, \underline{z}) \geq 0\} \ . \tag{5}$$

In this paper, we will only deal with tracking compact shapes. From this simplification, it follows that $\phi(\underline{x}_k^p, \underline{z})$ must have an upper bound. Then, we proceed to define $\phi_{max}(\underline{x}_k^p)$ as

$$\phi_{max}(\underline{x}^p) := \max_{\underline{z} \in \mathcal{S}(\underline{x}_k^p)} \ \phi(\underline{x}_k^p, \underline{z}) \ . \tag{6}$$

Definition 2 (Level-set)
For a given shape function $\phi(\underline{x}_k^s, \underline{z})$, a level-set $\mathcal{L}_\phi(c)$ is the region where the shape function takes the value of c,

$$\mathcal{L}_\phi(c) := \{\underline{z} \mid \phi(\underline{x}_k^p, \underline{z}) = c\} \ . \tag{7}$$

It can be seen that every point $\underline{z}$ belongs to a level-set. In particular, $\mathcal{S}(\underline{x}_k^p)$ can be seen as the union of all level-sets for $c \geq 0$. The target shape boundary is $\mathcal{L}_\phi(0)$. The calculation of level-sets is not straightforward, and a variety of modeling techniques exist [21] [22] [23]. However, Level-Set RHMs do not require an explicit calculation of these. Fig. 3 shows the level-sets of the shapes in Fig. 2 using the signed Euclidian distance, with the highlighted level-set $\mathcal{L}_\phi(0.3)$.

From this, a generative model for all points of $\mathcal{S}(\underline{x}_k^p)$ in function of the shape parameters $\underline{x}_k^p$ can be derived. First, a scalar c is selected from the interval $[0, \phi_{max}(\underline{x}_k^p)]$. Then, a measurement source $\underline{z}_{k,i}$ is chosen from the level-set $\mathcal{L}_\phi(c)$.

As the measurement source $\underline{z}_{k,i}$ is unknown, its corresponding scalar c is also unknown. Hence, it has to be modeled as a random variable $\boldsymbol{s}_{k,i}$, in a similar fashion to Example 1. In order to make $\boldsymbol{s}_{k,i}$ independent of the tracked shape, $\phi_{max}(\underline{x}_k^p)$ is used as a normalization factor. Then, for a given $s_{k,i}$ drawn from $\boldsymbol{s}_{k,i}$, the corresponding scaled shape is the level-set $\mathcal{L}_\phi(s_{k,i} \cdot \phi_{max}(\underline{x}_k^p))$.

Hence, it follows that if we model a measurement source $\underline{z}_{k,i}$ as being related to the scaling factor $s_{k,i}$, it holds that

$$\phi(\underline{x}_k^p, \underline{z}_{k,i}) = s_{k,i} \cdot \phi_{max}(\underline{x}_k^p) \ . \tag{8}$$

Finally, based on (1) and (8), the following measurement equation h is obtained,

$$\begin{aligned} 0 &= \phi(\underline{x}_k^p, \underline{z}_{k,i}) - s_{k,i} \cdot \phi_{max}(\underline{x}_k^p) \\ &= \phi(\underline{x}_k^p, \underline{y}_{k,i} - \underline{w}_{k,i}) - s_{k,i} \cdot \phi_{max}(\underline{x}_k^p) \\ &:= h(\underline{\boldsymbol{x}}_k, \underline{y}_{k,i}, \underline{\boldsymbol{w}}_{k,i}, \boldsymbol{s}_{k,i}) \ . \end{aligned} \tag{9}$$

In this way, the measurement, the scaling factor, and the target state are mapped to the pseudo-measurement 0.

IV. Level-Set RHMs using polygons

In this section, a Level-Set RHM will be 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 seen in Sec. IV-D, 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 $\underline{b}_{n-1}$ is also connected to $\underline{b}_0$. The points in the chain sequence will be referred as the *vertices* of the polygon.

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

$$\underline{x}_k^p = \left(\underline{b}_{k,0}^T, \ldots, \underline{b}_{k,n-1}^T\right)^T .$$

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

B. Shape Function

The shape function $\phi^\pi(\underline{x}_k^p, \underline{z})$ used in polygonal RHMs is the signed Euclidian distance. In order to describe it, first we need a function $d(\underline{x}_k^p, \underline{z})$ that returns the Euclidian distance between an arbitrary point $\underline{z}$ and the closest point in the boundary of the polygon $\mathcal{S}(\underline{x}_k^p)$. The closest point is often not unique, however the particular correspondence is not important, only the distance itself.

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

$$\text{clamp}(u) = \begin{cases} 0, & \text{if } u < 0 \\ u, & \text{if } 0 \leq u \leq 1 \\ 1, & \text{otherwise.} \end{cases} \tag{10}$$

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

$$u_i := \frac{(\underline{z} - \underline{b}_{k,i}) \cdot (\underline{b}_{k,i+1} - \underline{b}_{k,i})}{||\underline{b}_{k,i+1} - \underline{b}_i||^2} \quad \text{, and} \tag{11}$$

$$\underline{z}_i^\pi = \underline{b}_{k,i} + \text{clamp}(u_i) \cdot \left(\underline{b}_{k,i+1} - \underline{b}_{k,i}\right) \quad . \tag{12}$$

Finally, the distance of the polygon to $\underline{z}$ is simply

$$d(\underline{x}_k^p, \underline{z}) = \min_{i \in \{0, \ldots, n-1\}} ||\underline{z} - \underline{z}_i^\pi|| \quad . \tag{13}$$

Thus, the distance complexity is in $\mathcal{O}(n)$. Of course, this is only an example implementation, and no claim is made that this is the most efficient one.

The signed distance function follows from this, in the form of

$$\phi^\pi(\underline{x}_k^p, \underline{z}) := \begin{cases} d(\underline{x}_k^p, \underline{z}), & \text{if } \underline{z} \in \mathcal{S}(\underline{x}_k^p) \\ -d(\underline{x}_k^p, \underline{z}), & \text{otherwise.} \end{cases} \tag{14}$$

Checking if $\underline{z}$ is inside $\mathcal{S}(\underline{x}_k^p)$ can be done using the even-odd rule, winding numbers, or heuristics based on nearest edges. These algorithms are usually in $\mathcal{O}(n)$.

It can be seen that $\phi^\pi(\underline{x}_k^p, \underline{z})$ implicitly produces a polygon using Definition 1. The only remaining piece is the calculation of $\phi_{max}^\pi(\underline{x}_k^p)$. However, for polygons, it can be seen that the points with the maximum distance to the edges lie in the intersection between vertex angle bisectors. Thus, a simple (if crude) way to find $\phi_{max}^\pi(\underline{x}_k^p)$ is by checking all these intersections (i.e., $\mathcal{O}(n^2)$), and then pick the maximum distance. This yields a complexity of $\mathcal{O}(n^3)$. It should be pointed out that, for the commonly used sample-based Bayesian estimators, in cases of low state uncertainty, the value of $\phi_{max}^\pi(\underline{x}_k^p)$ will change very little between samples. In these cases, it makes sense to compute it only once per measurement in the update step.

C. Scaling Distribution

The choice of distribution for the scaling factor $\boldsymbol{s}$ varies depending on the application. A common example is when measurement sources are generated only from the edges. In these cases, as scaling begins from the edges, the only scaling is 0. This is an instance of *shape fitting*. In this way, fitting can be seen as a special case of Level-Set RHMs.

In other cases, it is only known how the measurements sources are generated in relation to the target shape, and not the shape function. For instance, in the common case of a flat surface being observed by a sensor, the measurement sources, represented as $\underline{\boldsymbol{z}}$, will usually be uniformly distributed in the target shape. Tight group targets where a minimum distance between participants must be held also come close to a uniform distribution. Thus, it makes sense to evaluate how the random variable $\boldsymbol{s}$, or in particular $\phi(\underline{x}_k^p, \underline{\boldsymbol{z}})$, behaves when $\underline{\boldsymbol{z}}$ is uniformly distributed. This then can be used as a starting point for other particular distributions of $\underline{\boldsymbol{z}}$.

First, the distribution of $\boldsymbol{s}$ with a circle and the signed Euclidian distance will be considered. In [5], it was shown that if measurement sources are uniformly distributed, and the scaling parameter $\boldsymbol{s}_c$ starts from the center, it follows that $\boldsymbol{s}_c^2$ is uniformly distributed as well. In this paper, the scaling starts from the border, and thus, $(1 - \boldsymbol{s})^2$ is uniform instead. It follows that, for circles, the distribution of $\boldsymbol{s}$ is

$$f^c(s) = \begin{cases} 2 - 2s & \text{for } s \in [0, 1] \\ 0 & \text{otherwise.} \end{cases} \tag{15}$$

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 Euclidean distances. Fig. 5 shows the results for an ellipse of aspect ratio 2 : 1, and the shapes introduced in Fig. 2. 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.

Calculating the exact distribution of $\boldsymbol{s}$ for a known shape is generally not efficient. It also happens that the exact target shape is not known, either. Thus, for a wide variety of shapes, a proper scaling factor for uniformly distributed measurement sources can be approximated from a uniformly distributed random variable $\boldsymbol{u} \sim \mathcal{U}(0, 1)$ in the form of

$$\boldsymbol{s} = 1 - \sqrt{\boldsymbol{u}} \quad , \tag{16}$$

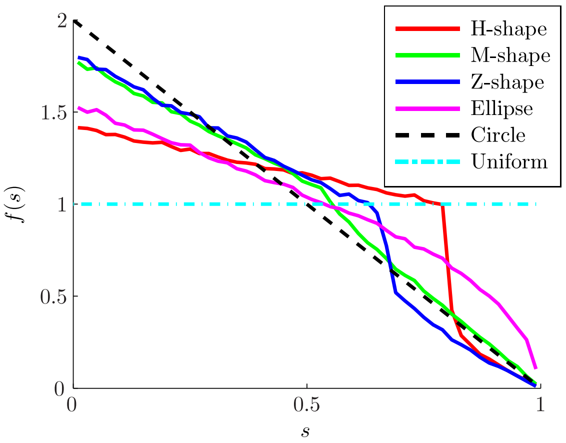


Figure 5: Distribution of scaling parameter.

while for extremely elongated shapes, a uniform distribution for $\boldsymbol{s}$ might be preferred instead.

D. Regularisation

When tracking a shape, 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. Even if the level of detail is known, a properly detailed shape model might fail to work in cases of high noise. In addition, even in cases of low noise, the shape may fail to be properly tracked if the initialization was unsuitable, causing the estimator to become stuck in a local minimum. A thorough discussion of the topic is outside the scope of this paper. In this section, a simple form of *regularization* will be presented.

In order to develop an approach to solve these issues, we followed some ideas of *Active Contour Models* [24], which faces 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 has the form

$$E_i = \int_0^1 \alpha(t) \left|\dot{\underline{\pi}}(t)\right|^2 + \frac{1}{2}\beta(t) \left|\ddot{\underline{\pi}}(t)\right|^2 dt \ , \tag{17}$$

where $\underline{\pi}(t)$ is the parametrization of a manifold for $t \in [0, 1]$, and $\dot{\underline{\pi}}(t)$ and $\ddot{\underline{\pi}}(t)$ are the corresponding first and second derivatives. Its task is to make the shape boundary *smoother* by reducing the stretch and curvature. 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 segments connecting to vertex $\underline{b}_{k,i}$ can be seen as springs, slightly pulling $\underline{b}_{k,i}$ towards $\underline{b}_{k,i-1}$ and $\underline{b}_{k,i+1}$. This pull is measured using the regularization factor $a_k \in \mathbb{R}$. Thus, the pull has the form

$$\underline{b}_{k+1,i} = (1 - 2a_k) \cdot \underline{b}_{k,i} + a_k \cdot (\underline{b}_{k,i-1} + \underline{b}_{k,i+1}) \ . \tag{18}$$

In effect, this ends up minimizing the term of the first derivative. A similar approach, not explored in this paper, can be used to reduce the term of the second derivative. This implementation has the extremely useful effect that the evolution in time is linear. Thus, it can be modeled using a matrix $\mathbf{H}(a_k)$, in the form

$$\underline{\boldsymbol{x}}^p_{k+1} = \mathbf{H}(a_k) \cdot \underline{\boldsymbol{x}}^p_k \ . \tag{19}$$

An implementation is therefore straightforward, including an analytical solution for the random vector $\underline{\boldsymbol{x}}^p_{k+1}$.

E. Bayesian Estimator

In order to track a target shape, the hidden state parameters must be inferred from the observed measurements. For this, a Gaussian-assumed Bayesian state estimator such as [19] can be used.

In these estimators, it is assumed that the state $\underline{\boldsymbol{x}}$ is a Markov process, and that $\underline{\boldsymbol{x}}_k$ is distributed as follows,

$$\underline{\boldsymbol{x}}_k \sim \mathcal{N}(\underline{x}; \hat{\underline{x}}_k, \mathbf{C}_{x_k}) \ . \tag{20}$$

Bayesian estimation includes a *prediction step* and a *filter step*. The prediction step evolves the system from an estimated time step k to a future time step $k+1$, according to the dynamic model. The filter step corrects the predicted state using observed measurements $\mathcal{Y}_k$, by applying a measurement function such as the one introduced in (9).

V. Evaluation

The evaluation of Level-Set RHMs consists of tracking the shapes introduced in Fig. 2. Two scenarios are explored. In the first one, the shape of a static target will be estimated. In the second scenario, the shape and position of a moving target will be tracked.

For backward inference, a sample-based Bayesian estimator is used. For this, the *Smart Sampling Kalman Filter* (S2KF) [19] was chosen. The benefit of the S2KF is its careful sample positioning, producing results closer to the analytic solutions. A small sampling factor of 2 times the number of dimensions is used. Of course, Level-Set RHMs can also be used with other sample-based filters. For each scenario, in each time step 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 $\mathbf{C}_{w_k}$. For $\boldsymbol{s}$, the approximation suggested in (16) is used.

An important part of the experiments is to measure how the estimated shape $\mathcal{S}(\underline{x}^p_k)$ converges to the target shape $\mathcal{S}_G$. Let the symmetric difference between both shapes be denoted as $\mathcal{S}(\underline{x}^p_k) \ \Delta \ \mathcal{S}_G$. In addition, let $|\mathcal{S}(\underline{x}^p_k) \ \Delta \ \mathcal{S}_G|$ 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 as follows,

$$\epsilon(k) := \frac{|\mathcal{S}(\underline{x}^p_k) \ \Delta \ \mathcal{S}_G|}{|\mathcal{S}_G|} \ . \tag{21}$$

For the measurement noise $\mathbf{C}_{w_k}$, 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 $10^{-2} \cdot \mathbf{I}$.

$\underline{\boldsymbol{x}}_k$ is, as mentioned before, assumed to be Gaussian. $\underline{\boldsymbol{x}}_0$ 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 will be referred as an n-polygon. Other state parameters, if present, are initialized with $\underline{0}$. The covariance matrix is initialized as $\mathbf{C}_{x_0} = 10^{-2} \cdot \mathbf{I}$.

A. 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. 5, the used distribution of $\boldsymbol{s}$ does not match the distribution of distances in the Z-shape. In order to compensate for this, a weak regularization factor of $a_k = 10^{-4}$ is used. A process noise of $10^{-5} \cdot \mathbf{I}$ is assumed at each time step. The target is tracked using a 6-polygon.

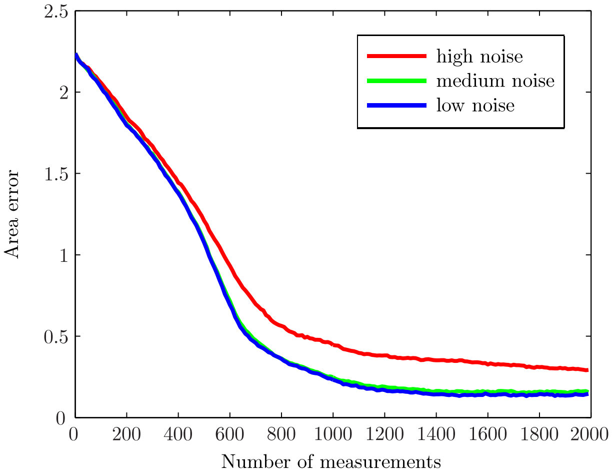


Figure 6: Area error while tracking static Z-shape.

Fig. 6 shows the averaged results of the area error for 20 runs. For low (blue) and medium noise (green), the shape has converged by 1100 measurements, with an area error of around 9% and 14% respectively. For high noise (red), the error remains relatively high between 40% and 50%. After that, at 1200 measurements, the error remains constant as a consequence of the assumptions done by the filter. On the one hand, the distribution of $\boldsymbol{s}$ is an approximation, and on the other hand the regularization procedure will always slightly pull the shape into a smoother, but incorrect, form. Thus, a closer convergence towards an error of 0% requires further adaptation of the regularization approach. Even so, the results are extremely close.

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. 10, Fig. 11, and Fig. 12. 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. 10 shows the results using high noise, Fig. 11 using medium noise, and Fig. 12 using low noise.

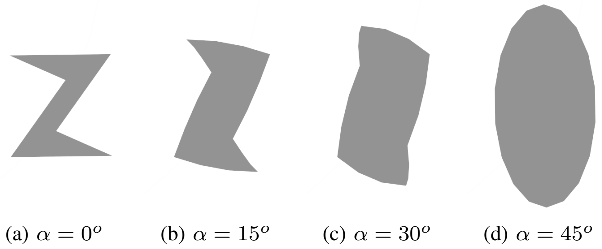

(a) $\alpha = 0^o$ (b) $\alpha = 15^o$ (c) $\alpha = 30^o$ (d) $\alpha = 45^o$

Figure 7: Morphing stages from Z-shape to ellipse.

B. 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. 8. It consists of the following parts.

- The shape rotates around a circle of radius 6 m, clockwise, starting in the positive x-axis.
- 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. 7 shows the morphing stages, which go on back and forth every 90^o.
- Every 45^o represents 1500 measurements.

A moderate regularization factor of $a_k = 6.3 \cdot 10^{-4}$ is used, together with a process noise of $3.16 \cdot 10^{-4} \cdot \mathbf{I}$. A 6-polygon was used to track the shape. The state parameters are extended with a constant velocity model. Note that the morphing is not an explicit part of the dynamic model, in order to explore the capabilities of our regularization approach from Sec. IV-D.

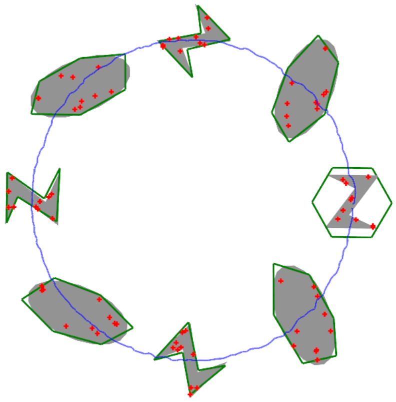


Figure 8: Experiment setup and results.

In Fig. 8, the estimated shapes are in green, the ground truth in gray, example measurements in red, and in blue the mean of the estimated polygon points. In Fig. 9 the average area errors for 30 runs in function of the angle are shown.

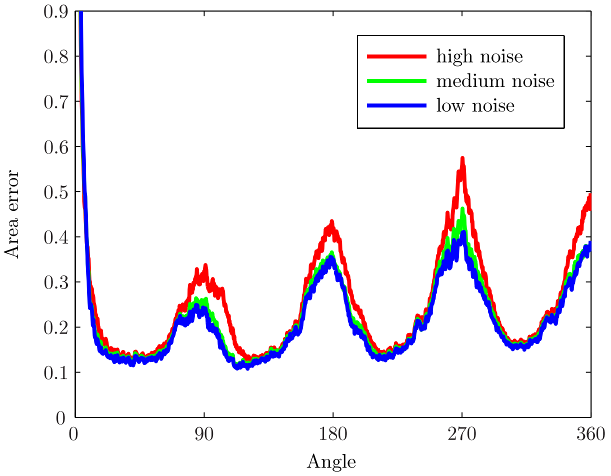


Figure 9: Area error while tracking morphing shape.

The morphing stages are evident. 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. 6, which 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.

VI. 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 shapes that are not star-convex. An implementation using polygons and regularization was described, and the evaluation showed the capability of approach to track complex, changing shapes in noisy scenarios.

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

[2] N. Chernov, *Circular and Linear Regression: Fitting Circles and Lines by Least Squares*. CRC Press, 2010.

[3] M. Baum and U. D. Hanebeck, “Extended Object Tracking with Random Hypersurface Models (to appear),” *IEEE Transactions on Aerospace and Electronic Systems*, accepted February 2013.

[4] F. Faion, M. Baum, and U. D. Hanebeck, “Tracking 3D Shapes in Noisy Point Clouds with Random Hypersurface Models,” in *Proceedings of the 15th International Conference on Information Fusion (Fusion 2012)*, Singapore, Jul. 2012.

[5] 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.

[6] 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, Jul. 2011.

[7] ——, “Random Hypersurface Models for Extended Object Tracking,” in *Proceedings of the 9th IEEE International Symposium on Signal Processing and Information Technology (ISSPIT 2009)*, Ajman, United Arab Emirates, Dec. 2009.

[8] 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, October 2005.

[9] 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.

[10] N. Petrov, L. Mihaylova, A. Gning, and D. Angelova, “Group Object Tracking with a Sequential Monte Carlo Method Based on a Parameterised Likelihood Function,” *Monte Carlo Methods and Applications*, 2012.

[11] N. Petrov, A. Gning, L. Mihaylova, and D. Angelova, “Box Particle Filtering for Extended Object Tracking,” in *15th International Conference on Information Fusion (FUSION)*. IEEE, 2012, pp. 82–89.

[12] M. Feldmann and D. Fränken, “Tracking of Extended Objects and Group Targets using Random Matrices – A New Approach,” *Proceedings of the 11th International Conference on Information Fusion (Fusion 2008)*, pp. 1–8, July 2008.

[13] ——, “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, Washington, July 2009.

[14] M. Feldmann, D. Fränken, and W. Koch, “Tracking of Extended Objects and Group Targets using Random Matrices,” *IEEE Transactions on Signal Processing*, vol. 59, no. 4, pp. 1409 –1420, 2011.

[15] U. Orguner, “A Variational Measurement Update for Extended Target Tracking With Random Matrices,” *IEEE Transactions on Signal Processing*, vol. 60, no. 7, pp. 3827 –3834, July 2012.

[16] K. Granström and U. Orguner, “A PHD Filter for Tracking Multiple Extended Targets Using Random Matrices,” *IEEE Transactions on Signal Processing*, vol. 60, no. 11, pp. 5657 –5671, nov. 2012.

[17] J. Lan and X. Rong Li, “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, pp. 2177 –2184.

[18] ——, “Tracking of Extended Object or Target Group Using Random Matrix -Part II: Irregular Object,” in *15th International Conference on Information Fusion (FUSION)*, july 2012, pp. 2185 –2192.

[19] U. D. Hanebeck and J. Steinbring, “Progressive Gaussian Filtering Based on Dirac Mixture Approximations,” in *Proceedings of the 15th International Conference on Information Fusion (Fusion 2012)*, Singapore, Jul. 2012.

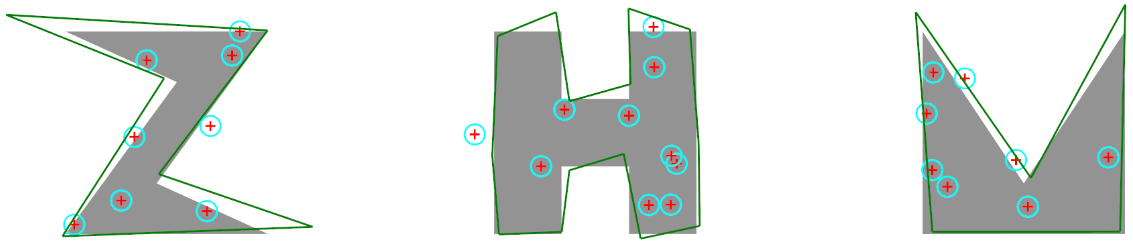
[20] M. Baum and U. D. Hanebeck, “Tracking an Extended Object Modeled as an Axis-Aligned Rectangle,” in *4th German Workshop on Sensor Data Fusion: Trends, Solutions, Applications (SDF 2009), 39th Annual Conference of the Gesellschaft für Informatik e.V. (GI)*, Lübeck, Germany, Oct. 2009.

[21] S. Osher and R. Fedkiw, *Level set methods and dynamic implicit surfaces*. Springer, 2002, vol. 153.

[22] S. Osher and N. Paragios, *Geometric level set methods in imaging, vision, and graphics*. Springer, 2003.

[23] R. Malladi, J. Sethian, and B. Vemuri, “Shape modeling with front propagation: a level set approach,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 17, no. 2, pp. 158–175, Feb.

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

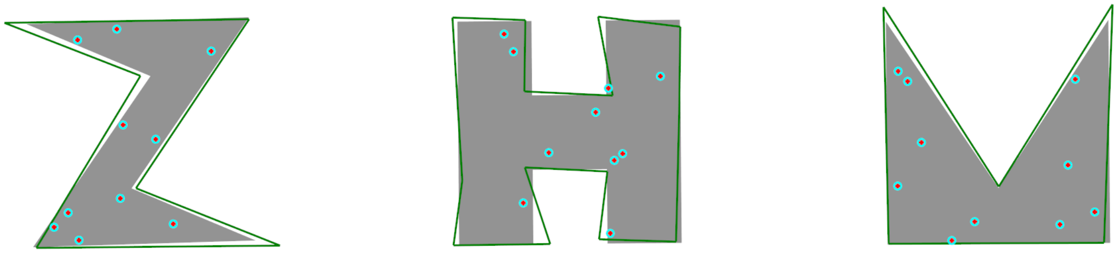


(a) Z-shape, $\epsilon = 40\%$

(b) H-shape, $\epsilon = 18\%$

(c) M-shape, $\epsilon = 23\%$

Figure 10: Tracking shape with high measurement noise.

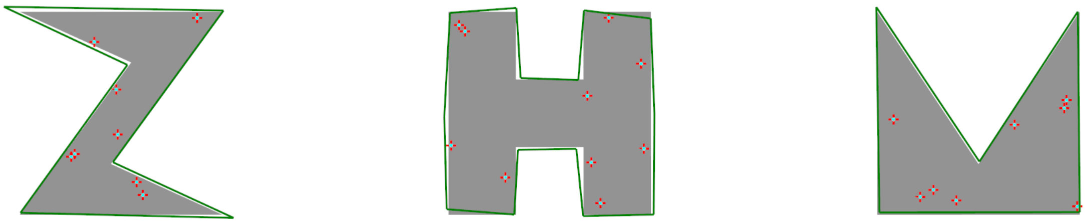


(a) Z-shape, $\epsilon = 15\%$

(b) M-shape, $\epsilon = 9.5\%$

(c) M-shape, $\epsilon = 9.0\%$

Figure 11: Tracking shape with medium measurement noise.



(a) Z-shape, $\epsilon = 9.2\%$

(b) M-shape, $\epsilon = 6.9\%$

(c) M-shape, $\epsilon = 5.7\%$

Figure 12: Tracking shape with low measurement noise.