Computational Statistics and Data Analysis 102 (2016) 1–22

Contents lists available at ScienceDirect

Computational Statistics and Data Analysis

journal homepage: www.elsevier.com/locate/csda

Estimation of linear target-layer trajectories using cluttered point cloud data

Darshan Bryner [a,*], Fred Huffer [b], Michael Rosenthal [a], J. Derek Tucker [c], Anuj Srivastava [b]

[a] Naval Surface Warfare Center, Panama City Division - X13, 110 Vernon Avenue, Panama City, FL 32407-7001, United States
[b] Department of Statistics, Florida State University, Tallahassee, FL 32306, United States
[c] Sandia National Laboratories, PO Box 5800 MS 1202, Albuquerque, NM 87185, United States

ARTICLE INFO

Article history:
Received 7 May 2015
Received in revised form 31 March 2016
Accepted 1 April 2016
Available online 7 April 2016

Keywords:
Spatial point process
Poisson process
Point clouds
Line detection

ABSTRACT

The problem of estimating a target-layer trajectory, modeled by a straight line, in 2D point clouds that contain target locations and overwhelming clutter is studied. These point clouds are generated by an image-based pre-processing tool, termed ATR, operating on SONAR image data that results in: (1) point locations and (2) an ATR score: a measure of the "target-likeness" for each point. The model of choice assumes that the observed point cloud is a superposition of two spatial processes: (1) a 1D Poisson process along the target-layer line, corrupted by 2D Gaussian noise, denoting target locations and (2) a 2D Poisson process denoting clutter. It is further assumed that the target-likeness measure follows known probability distributions for both target locations and clutter. The line is parameterized by distance from the origin and the angle with respect to a horizontal axis, and the likelihood of these parameters for observed data is derived. Using a maximum-likelihood approach, a gradient-based estimate for line parameters and other nuisance parameters is developed. A formal procedure that tests for the presence of a target-layer trajectory in the point cloud data is additionally developed. The success of this method in both simulated and real datasets collected by NSWC PCD is demonstrated.

Published by Elsevier B.V.

1. Introduction

The problem of detecting targets – both underwater and overground – is important in both civilian and military contexts. The term target is used to describe an object of interest in a scene that one would like to locate. The process of locating underwater targets has military applications in mine countermeasures, but it can also be directed to the detection of other more general important objects such as lost cargo or debris from a sunken vessel or aircraft. In the context of mine countermeasures, targets are unspent explosive devices that have been left during wartime efforts in strategic marine locations that are potentially catastrophic for transiting military, passenger, and merchant ships. In the context of recovering precious cargo, targets may include anything. In the case of a downed airplane resulting in broken and scattered equipment, it is often important to recover as much of the plane as possible so that investigators can determine what caused the crash. In this paper we focus on the problem of detecting underwater targets that are distributed along trajectories. The main

* Corresponding author. Tel.: +1 850 230 7045; fax: +1 850 235 5374.
E-mail addresses: darshan.bryner@navy.mil (D. Bryner), huffer@stat.fsu.edu (F. Huffer), michael.m.rosenthal@navy.mil (M. Rosenthal), jdtuck@sandia.gov (J.D. Tucker), anuj@stat.fsu.edu (A. Srivastava).

http://dx.doi.org/10.1016/j.csda.2016.04.002
0167-9473/Published by Elsevier B.V.

sensor for detecting underwater targets is side-scan SONAR, a (sound-based) imaging device that scans the ocean floor and generates image maps of the observed terrain (Chandra et al., 2002). A SONAR device emits sound waves in different audio frequencies and measures the waves reflected/scattered by different objects in an observation space; the strengths of these returns form pixel values in resulting images. Image analysts design algorithms that study patterns of pixels in these images to detect the resemblance of targets amongst a tremendous amount of natural and artificial debris that is littered across the ocean floor. The approaches/algorithms for detecting target occurrences using image pixels are generically termed as *automated target recognition* (ATR) (Chandra et al., 2002). Several ATR procedures with varying degrees of success have been proposed over the years, relying on ideas ranging from signal processing, machine learning, and statistical modeling (see Stack, 2011 and Ratches, 2011 for description of the state of the art). The general goal of ATR algorithms is to detect, recognize, and help neutralize targets using SONAR images. In most general situations the current ATR performance remains mediocre and one requires additional statistical modeling to achieve further improvements in target detection performance.

If we focus only on coordinates identified by an ATR algorithm as potential target locations, we obtain a 2D point cloud in the region of interest. This assumption of point data is justified by the fact that the spatial extent of most targets in SONAR images is typically only a few pixels – ranging from tens to hundreds – due to the small size of targets relative to the bandwidths used in synthetic aperture SONAR (SAS) imaging. Thus, the limited spatial extent of targets allows us to treat them as individual points in the observed spatial domain, and the focus shifts from appearance-based pixel patterns to location-based spatial patterns of target locations. We will also assume the availability of an *ATR score*, a real-number associated with each point, that quantifies the confidence an ATR algorithm has about the presence of a target at that point. The higher the number, the more likely it is a target.

This labeled 2D point cloud does not consist of target locations only. The main challenge in target detection comes from *target-like* objects that are present in imaged areas but are not targets. These include artificial debris (bottles, boxes, fish traps, etc.) and natural objects (fish, rock, coral, etc.) that have appearances (pixel values, object size, and pixel patterns) similar to targets in SONAR images. Since ATR algorithms rely on pixel patterns to perform target detection and classification, this often leads to an ATR algorithm generating a large number of false detections over the search space, and it becomes difficult to distinguish targets from these false detections, also termed *clutter*. Thus, these point clouds are heavily cluttered with the clutter points far outnumbering the target locations.

If we assume that targets are laid by a single target-layer (vessel), then the knowledge of target-layer trajectory can help discriminate between targets and clutter. For instance, one can focus only on detections that are reasonably close to the target-layer trajectory for identifying potential targets. Points that are far away can be easily discarded as clutter. Thus, the problem of estimation of *target-layer trajectory* becomes an important step in target detection. A target-layer trajectory, or simply a trajectory, is potentially any smooth curve in $\mathbb{R}^2$, and estimating it without any additional information is quite difficult. Note that the detected points do not have any time or sequence information associated with them. The space of all smooth curves is infinite-dimensional and requires additional constraints to make the problem tractable. One solution is to assume a simple geometry associated with the trajectory, such as a line or a quadratic, and restrict the search space to the relatively small number of parameters characterizing that geometry. In this paper we assume that the target-layer trajectories are straight-lines and focus on estimating their occurrences in the observed scene. The line estimation is performed using spatial point data where each point represents a potential target location detected using an ATR algorithm.

There have been several efforts that use a point-process model for target detection. Most of them directly focus on point process models for target detection Agarwal et al. (2002), Lake (1998), Cressie and Lawson (1998), Trang et al. (2008), Trang et al. (2011), Bryner et al. (2014) and Walsh and Raftery (2005), while some try this approach implicitly using clutter removal Byers and Raftery (1998). The geometry of the underlying target-layer trajectory is seldom exploited in these papers. One exception is Walsh and Raftery (2002), which assumes target locations are nearly linear and uses that extra knowledge to help improve target estimation. Another example is Lake et al. (1997), which assumes collinearity of target locations to perform target detection. In contrast to these papers, we will focus on the trajectory itself and will estimate its parameters assuming a single straight-line model.

We will initially assume that the observed points are of two types: (1) **targets**: a set of points representing objects of interest which are the points from a 1D Poisson process along the target-layer line randomly displaced by Gaussian noise with unknown variance, and (2) **clutter**: a set of points arising from a 2D Poisson process on the observation domain. The two processes are assumed to be independent of each other. The full observation process is a union of these two sets. Associated with each point is an ATR score that reflects its likeness to a target. Taking a maximum-likelihood approach we derive a likelihood function for a line model given the full set of observations, and maximize it using a gradient approach. This framework is very similar to the one used in Su et al. (2013) for estimating arbitrary curves (with known shapes) in a cluttered point cloud. The main difference lies in the assumption of a straight-line trajectory and the availability of ATR scores. This changes the likelihood functions and allows for a direct estimation of a line guided by the ATR scores. Later on we extend the model to include: (1) estimation of multiple linear trajectories, and (2) presence of targets that are not associated with target-layer trajectories but are scattered according to an independent Poisson process in the observed region.

The rest of the paper is arranged as follows. We start by describing the observation model and the associated likelihood function, without the ATR scores, in Section 2, and expand the framework to include the ATR scores in Section 3. We then present estimation results on three types of datasets in Section 4. Section 5 compares the performance of our line model with RANSAC, a widely used algorithm for detecting lines or curves in point cloud data. Section 6 outlines and provides

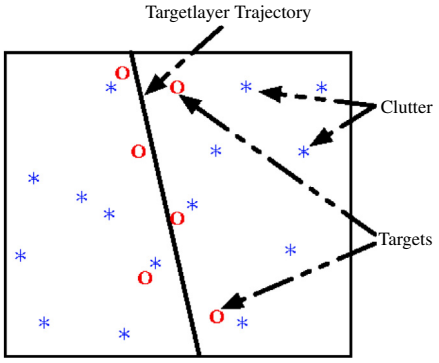

Fig. 1. A display of the model behind the observed point cloud. The points associated with the target-layer trajectory (dark line) are shown as red 'o' while the background clutter is shown using blue '*'. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

examples of a formal procedure that tests the statistical significance of the fitted line model against a homogeneous null model. We give some brief concluding discussion in Section 7.

2. Line detection model using point data

Let the region of interest for target detection be denoted by $U \subset \mathbb{R}^2$. If there are targets in U, we believe they arose by a target-layer making a single pass over the region, following roughly a straight line, and scattering targets about this line. The ATR pre-processing results in m locations $\boldsymbol{Y} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_m\}$ in U. Each location denotes either a target or clutter. In the first step, we will ignore the ATR scores and treat all points as equal in the analysis. Later on, we will include these scores in the likelihood function and study their influence in line estimation.

Our goal is to formulate a model for the locations of targets and clutter points, estimate the parameters of this model, and use this model to test the hypothesis that targets are present. We treat this as a binary hypothesis test—the null hypothesis is that $\boldsymbol{Y}$ is simply clutter, and the alternate hypothesis is that $\boldsymbol{Y}$ is a superposition of targets scattered about some unknown line L and clutter. If $\boldsymbol{Y}$ contains more points lying closer to some line than would be expected by chance under our model, we reject the null hypothesis that all the points are clutter and conclude that a target line is present (see Fig. 1).

We now present the stochastic model behind the point observations.

1. **Target points**: The points associated with targets arise as follows. Random points $\boldsymbol{s}$ are generated on the line L by a Poisson process with intensity $\gamma(\boldsymbol{s})$, $\boldsymbol{s} \in L$. Each point $\boldsymbol{s}$ from this process is then randomly displaced in $\mathbb{R}^2$ according to the density $f(\boldsymbol{y}|\boldsymbol{s})$ on $\mathbb{R}^2$, which yields a point $\boldsymbol{y}$ that represents the location of a target. The collection of these randomly displaced points $\boldsymbol{y}$ forms a Poisson process on $\mathbb{R}^2$ with intensity $\rho(\boldsymbol{y}) = \int_L f(\boldsymbol{y}|\boldsymbol{s})\gamma(\boldsymbol{s})\, d\boldsymbol{s}$, $y \in \mathbb{R}^2$, where the integration is over the line L, and $d\boldsymbol{s}$ denotes Lebesgue measure on L. We restrict our attention to the points lying in the region U. The total number of targets in U will be a Poisson random variable with mean $\Gamma = \int_U \rho(\boldsymbol{y})\, d\boldsymbol{y}$, where the integration is over U and $d\boldsymbol{y}$ is Lebesgue measure on $\mathbb{R}^2$.
2. **Clutter points**: The points associated with clutter are modeled as a realization of a Poisson process with intensity $\lambda(\boldsymbol{y})$ on U. The number of clutter points is a Poisson variable with mean $\Lambda = \int_U \lambda(\boldsymbol{y})\, d\boldsymbol{y}$.

We assume the Poisson processes for targets and clutter are independent, and thus, their superposition is a Poisson process with intensity $\xi(\boldsymbol{y}) = \lambda(\boldsymbol{y}) + \rho(\boldsymbol{y})$. The point cloud $\boldsymbol{Y}$ is a realization from this process. The density of the Poisson process $\boldsymbol{Y}$ is $P(\boldsymbol{Y}) = e^{-\Lambda-\Gamma} \prod_{i=1}^{n(\boldsymbol{Y})} \xi(\boldsymbol{y}_i)$, where $n(\boldsymbol{Y})$ denotes the cardinality of $\boldsymbol{Y}$. The null hypothesis is that all the points are clutter, i.e. that $\rho(\boldsymbol{y}) \equiv 0$ and the density of $\boldsymbol{Y}$ is $Q(\boldsymbol{Y}) = e^{-\Lambda} \prod_{i=1}^{n(\boldsymbol{Y})} \lambda(\boldsymbol{y}_i)$.

The line L, the intensities $\gamma(\cdot)$ and $\lambda(\cdot)$, and the density $f(\boldsymbol{y}|\boldsymbol{s})$ will be described (see below) in terms of parameters whose values are not known *a priori* and must be estimated from the data. Therefore, in our hypothesis testing we will use the generalized likelihood ratio test (GLRT). The GLRT statistic is given by

$$\frac{\max_{\boldsymbol{\theta_0}} Q(\boldsymbol{Y}|\boldsymbol{\theta_0})}{\max_{\boldsymbol{\theta}} P(\boldsymbol{Y}|\boldsymbol{\theta})}, \tag{1}$$

where $\boldsymbol{\theta_0}$ and $\boldsymbol{\theta}$ denote the parameters involved in the null and alternative hypotheses, respectively. We will make the following assumptions:

1. The density $f(\boldsymbol{y}|\boldsymbol{s})$ is bivariate normal with mean $\boldsymbol{s}$ and variance $\sigma^2 I$, i.e., the points $y \in \boldsymbol{Y}$ corresponding to targets are obtained by adding i.i.d. $N(0, \sigma^2 I)$ noise to the points $\boldsymbol{s}$ sampled from L.
2. The intensities $\gamma(\boldsymbol{s})$ and $\lambda(\boldsymbol{y})$ are constant: $\gamma(\boldsymbol{s}) = \gamma$ for $\boldsymbol{s} \in L$ and $\lambda(\boldsymbol{y}) = \lambda$ for $\boldsymbol{y} \in U$.

With these assumptions, we have

$$\rho(\boldsymbol{y}) = \gamma\alpha_\sigma(\boldsymbol{y}), \quad \text{where } \alpha_\sigma(\boldsymbol{y}) = \frac{\exp\left(-d(\boldsymbol{y})^2/(2\sigma^2)\right)}{\sigma\sqrt{2\pi}}, \tag{2}$$

and $d(\boldsymbol{y})$ is the minimum distance between the point $\boldsymbol{y}$ and L. Also, $\xi(\boldsymbol{y}) = \lambda + \gamma\alpha_\sigma(\boldsymbol{y})$, $\Gamma = \gamma J$ where $J = \int_U \alpha_\sigma(\boldsymbol{y})\, d\boldsymbol{y}$, and $\Lambda = \lambda A$ where A is the area of U. We will call this our **baseline model**.

Appendix gives a derivation of (2) and formulas for computing the integral J when the region U is polygonal. Note that $\alpha_\sigma(\boldsymbol{y})$ is large if a point $\boldsymbol{y}$ is close to L, with the closeness measured relative to the scale σ. If σ is fairly small and the line L stays away from the boundary of U, we can approximate the integral J by the length of the segment $L \cap U$.

The line L will be characterized by the polar coordinates (r, ϕ) of the point on the line that is closest to the origin; L is the line tangent to the circle of radius r (centered at the origin) at the point (r, ϕ). Note that $\alpha_\sigma(\boldsymbol{y})$ in (2) and the integral J given above depend implicitly on the line L, and thus they are functions of (r, ϕ, σ). Define the vectors $\boldsymbol{v} = (\cos(\phi), \sin(\phi))$ and $\boldsymbol{w} = (-\sin(\phi), \cos(\phi))$. Here, $\boldsymbol{v}$ is a unit vector perpendicular to L, and $\boldsymbol{w}$ is a unit vector parallel to L. Let $p(\boldsymbol{y}) = \boldsymbol{v} \cdot \boldsymbol{y} - r$ be the *signed* distance from the point $\boldsymbol{y}$ to L so that $d(\boldsymbol{y}) = |p(\boldsymbol{y})|$.

Define H to be the logarithm of $P(\boldsymbol{Y}|\lambda, \gamma, r, \phi, \sigma)$, and let $\boldsymbol{\theta} = [\lambda, \gamma, r, \phi, \sigma] \in \mathbb{R}^5$ denote the unknown parameters. The function $H : \mathbb{R}^5 \to \mathbb{R}$ is then given by

$$H(\boldsymbol{\theta}) = -\gamma J - \lambda A + \sum_{i=1}^{m} \log(\lambda + \gamma\alpha_\sigma(\boldsymbol{y}_i)),$$

and let $\hat{\boldsymbol{\theta}} = \text{argmax}_{\boldsymbol{\theta}} H(\boldsymbol{\theta})$ be the maximizer. We will solve for the MLE $\hat{\boldsymbol{\theta}}$ using a gradient approach. The derivatives of H with respect to these five parameters are given by:

$$\begin{aligned}
\frac{\partial H}{\partial \lambda} &= -A + \sum_{i=1}^{m} \frac{1}{\lambda + \gamma\alpha_\sigma(\boldsymbol{y}_i)}, \\
\frac{\partial H}{\partial \gamma} &= -J + \sum_{i=1}^{m} \frac{\alpha_\sigma(\boldsymbol{y}_i)}{\lambda + \gamma\alpha_\sigma(\boldsymbol{y}_i)}, \\
\frac{\partial H}{\partial r} &= -\gamma\frac{\partial J}{\partial r} + \sum_{i=1}^{m} \frac{\gamma\alpha_\sigma(\boldsymbol{y}_i)}{\lambda + \gamma\alpha_\sigma(\boldsymbol{y}_i)} \left(\frac{p(\boldsymbol{y}_i)}{\sigma^2}\right), \\
\frac{\partial H}{\partial \phi} &= -\gamma\frac{\partial J}{\partial \phi} + \sum_{i=1}^{m} \frac{\gamma\alpha_\sigma(\boldsymbol{y}_i)}{\lambda + \gamma\alpha_\sigma(\boldsymbol{y}_i)} \left(\frac{-p(\boldsymbol{y}_i)(\boldsymbol{w} \cdot \boldsymbol{y}_i)}{\sigma^2}\right), \\
\frac{\partial H}{\partial \sigma} &= -\gamma\frac{\partial J}{\partial \sigma} + \frac{\gamma}{\sigma} \sum_{i=1}^{m} \frac{\alpha_\sigma(\boldsymbol{y}_i)}{\lambda + \gamma\alpha_\sigma(\boldsymbol{y}_i)} \left(\frac{p(\boldsymbol{y}_i)^2}{\sigma^2} - 1\right).
\end{aligned}$$

Formulas for the partial derivatives of J are given in Appendix.

With regard to carrying out the GLRT, our situation is non-standard. Firstly, under the null hypothesis $\gamma = 0$ that $\mathbf{Y}$ is simply clutter, our model does not satisfy the usual regularity conditions required for validity of the asymptotic chi-squared distribution of $-2\log(\text{GLRT})$, where GLRT is given in (1). In particular, under the null it is easily seen that $\frac{\partial H}{\partial r}$, $\frac{\partial H}{\partial \phi}$, and $\frac{\partial H}{\partial \sigma}$ are identically zero so that the Fisher information matrix is singular. Secondly, if no restrictions are placed on the value of σ, the $\max_\sigma$ in the denominator of (1) is infinite. This follows by noting that, if λ and γ are fixed at any positive values, and r and ϕ are chosen so that some data point $\mathbf{y}_i$ lies exactly on the line L, then $H(\theta) \to \infty$ as $\sigma \to 0$. For these reasons, when carrying out the GLRT we rely on critical points determined by Monte Carlo and, moreover, restrict the allowed values of σ by requiring $\sigma \geq \sigma_0$ for some appropriately chosen value σ_0. This last restriction is reasonable in our application since there are technological limits (e.g., GPS location errors) on the accuracy of target locations.

The parameters involved in the null and alternative hypotheses are $\boldsymbol{\theta_0} = \lambda$ and $\boldsymbol{\theta} = (\lambda, \gamma, r, \phi, \sigma)$ with $\sigma \geq \sigma_0$. Using the assumptions and facts stated above, the GLRT simplifies to

$$\frac{\max_{\lambda} Q(\boldsymbol{Y}|\lambda)}{\max_{\lambda,\gamma,r,\phi,\sigma} P(\boldsymbol{Y}|\lambda, \gamma, r, \phi, \sigma)} = \frac{\max_{\lambda}\left(e^{-\lambda A}\prod_{i=1}^{m}\lambda\right)}{\max_{\lambda,\gamma,r,\phi,\sigma}\left(e^{-\gamma J-\lambda A}\prod_{i=1}^{m}(\lambda + \gamma\alpha_\sigma(\boldsymbol{y}_i))\right)}.$$

The numerator maximization is easily seen to be $e^{-m}(m/A)^m$. Finding the denominator requires numerical optimization.

We use the *fmincon* function in MATLAB to perform a gradient-based, constrained optimization of the function H, where our only constraint is that $\sigma \geq \sigma_0$. (The value of $\sigma_0 = 10^{-5}$ throughout this paper.) Since a gradient approach provides only a locally-optimal solution, we use multiple initializations of the line to search a larger space. We retain all solutions resulting from different initializations and select the best amongst them according to the likelihood function. Fig. 2 shows

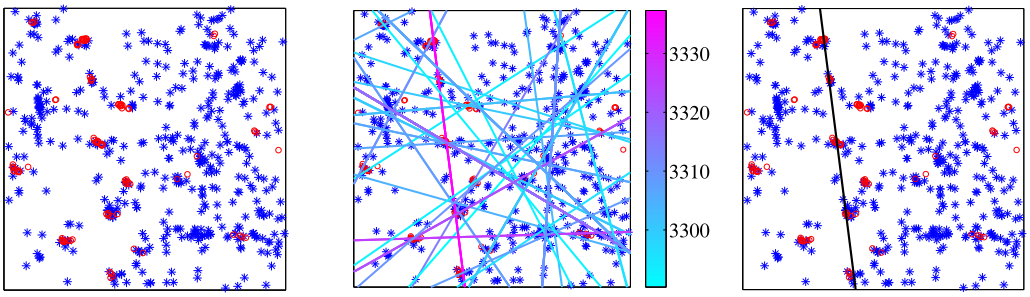

Fig. 2. Estimation of target-layer trajectory using maximum-likelihood estimation. The left panel shows the point observations, where red circles indicate ground truth target locations and blue stars indicate clutter. The center panel shows local solutions resulting from different initializations of the line, colorized according to the H value. The right panel shows the best solution amongst these local results. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

an illustration of this idea using a dataset from the Naval Surface Warfare Center, Panama City Division (NSWC PCD). See Section 4 for a detailed description of the data. The left panel of the figure shows the data, where red circles indicate ground truth target locations and blue stars indicate clutter points. The center panel shows the resulting lines obtained from the gradient-based optimization over θ for many different initial conditions, each colorized according to the associated (locally-optimal) H value. The right panel shows only the line with the highest H value from the center panel. Although there is no ground truth line available to validate this result, the best estimated line seems reasonable. For this dataset $m = 608$, $U = [0, 1] \times [0, 1]$, and the three parameters not associated with the line are estimated to be $\hat{\lambda} = 551.2$, $\hat{\gamma} = 56.26$, and $\hat{\sigma} = 0.008241$.

3. Line detection model with ATR scores and multiple trajectories

3.1. Extensions

The baseline model of Section 2 can be extended in the following three ways:

1. **Incorporate ATR scores**: Firstly, we consider the situation where each observed point location $\boldsymbol{y}_i$ has associated to it an ATR score $c_i \in \mathbb{R}_+$, and thus, our set of data consists of the m pairs $(\boldsymbol{y}_1, c_1), \ldots, (\boldsymbol{y}_m, c_m) \in U \times \mathbb{R}_+$. We assume as before that the targets and clutter occur according to independent Poisson processes with intensities $\rho(\boldsymbol{y})$ and $\lambda(\boldsymbol{y})$, respectively. In addition, we assume that the ATR scores for targets and clutter are independent of their locations, with scores for targets being *i.i.d.* from the density g_1, and scores for clutter *i.i.d.* from the density g_0. Then, the pairs $(\boldsymbol{y}, c)$ for targets form a marked Poisson process with intensity $\rho(\boldsymbol{y})g_1(c)$, and the pairs for clutter objects form a marked Poisson process with intensity $\lambda(\boldsymbol{y})g_0(c)$. (Here we regard the locations $\boldsymbol{y}$ as "points" and the scores c as "marks".) Our data is the superposition of these two independent marked Poisson processes, which is itself a marked Poisson process with intensity $\xi(\boldsymbol{y}, c) \equiv \lambda(\boldsymbol{y})g_0(c) + \rho(\boldsymbol{y})g_1(c)$. We will call this the *baseline model with ATR* scores.
2. **Allow multiple target-layer trajectories**: Secondly, we extend the model to allow multiple target-layer trajectories. We suppose there are k such trajectories following lines $L_1, \ldots, L_k$, each described by its own vector of parameters $(\gamma_j, r_j, \phi_j, \sigma_j), j = 1, \ldots, k$, with interpretations identical to those in the single trajectory case of Section 2. The target locations are now a superposition of k Poisson processes arising from the k trajectories, and the overall intensity of the target locations is
$$\rho(\boldsymbol{y}) = \sum_{j=1}^{k} \gamma_j \alpha_j(\boldsymbol{y}), \quad \text{where } \alpha_j(\boldsymbol{y}) = \frac{\exp\left(-d_j(\boldsymbol{y})^2/(2\sigma_j^2)\right)}{\sigma_j\sqrt{2\pi}},$$
and $d_j(\boldsymbol{y})$ is the minimum distance between the point $\boldsymbol{y}$ and the line L_j.
3. **Inhomogeneous clutter and background target process**: Thirdly, we extend the model to allow for a (possibly) non-constant clutter intensity $\lambda(\boldsymbol{y}) = \lambda f_0(\boldsymbol{y})$ where f_0 is some given intensity function. Finally, in some cases we shall extend the model to allow the presence of some targets that do not arise from a linear target-layer trajectory, but rather from a background target process that is an independent Poisson process with intensity $\beta f_1(\boldsymbol{y})$ for some given function f_1. This background process could represent previously placed targets in the region or some targets being placed in a field-like arrangement rather than along a linear trajectory.

With these extensions, our marked Poisson process has an intensity function of the form

$$\xi(\boldsymbol{y}, c) = \lambda f_0(\boldsymbol{y})g_0(c) + \beta f_1(\boldsymbol{y})g_1(c) + \sum_{j=1}^{k} \gamma_j \alpha_j(\boldsymbol{y})g_1(c), \tag{3}$$

and the data $(\boldsymbol{y}_1, c_1), \ldots, (\boldsymbol{y}_m, c_m)$ has a density of

$$e^{-\lambda A - \beta B - \sum_j \gamma_j J_j} \prod_{i=1}^{m} \xi(\boldsymbol{y}_i, c_i), \tag{4}$$

where $A = \int_U f_0(\boldsymbol{y})\, d\boldsymbol{y}$, $B = \int_U f_1(\boldsymbol{y})\, d\boldsymbol{y}$ and $J_j = \int_U \alpha_j(\boldsymbol{y})\, d\boldsymbol{y}$, $j = 1, \ldots, k$.

We solve for the MLE by maximizing the logarithm of this density

$$H(\boldsymbol{\theta}) = -\lambda A - \beta B - \sum_{j=1}^{k} \gamma_j J_j + \sum_{i=1}^{m} \log \xi(\boldsymbol{y}_i, c_i),$$

where $\boldsymbol{\theta}$ denotes the vector of all the parameters in our model: λ, β, and $(\gamma_j, r_j, \phi_j, \sigma_j), j = 1, \ldots, k$. The partial derivatives of H are given by

$$\begin{aligned}
\frac{\partial H}{\partial \lambda} &= -A + \sum_{i=1}^{m} \frac{f_0(\boldsymbol{y}_i) g_0(c_i)}{\xi(\boldsymbol{y}_i, c_i)}, \\
\frac{\partial H}{\partial \beta} &= -B + \sum_{i=1}^{m} \frac{f_1(\boldsymbol{y}_i) g_1(c_i)}{\xi(\boldsymbol{y}_i, c_i)}, \\
\frac{\partial H}{\partial \gamma_j} &= -J_j + \sum_{i=1}^{m} \frac{\alpha_j(\boldsymbol{y}_i) g_1(c_i)}{\xi(\boldsymbol{y}_i, c_i)}, \\
\frac{\partial H}{\partial r_j} &= -\gamma_j \frac{\partial J_j}{\partial r_j} + \sum_{i=1}^{m} \frac{\gamma_j \alpha_j(\boldsymbol{y}_i) g_1(c_i)}{\xi(\boldsymbol{y}_i, c_i)} \left(\frac{p_j(\boldsymbol{y}_i)}{\sigma_j^2} \right), \\
\frac{\partial H}{\partial \phi_j} &= -\gamma_j \frac{\partial J_j}{\partial \phi_j} + \sum_{i=1}^{m} \frac{\gamma_j \alpha_j(\boldsymbol{y}_i) g_1(c_i)}{\xi(\boldsymbol{y}_i, c_i)} \left(\frac{-p_j(\boldsymbol{y}_i)(\boldsymbol{w}_j \cdot \boldsymbol{y}_i)}{\sigma_j^2} \right), \\
\frac{\partial H}{\partial \sigma_j} &= -\gamma_j \frac{\partial J_j}{\partial \sigma_j} + \frac{\gamma_j}{\sigma_j} \sum_{i=1}^{m} \frac{\alpha_j(\boldsymbol{y}_i) g_1(c_i)}{\xi(\boldsymbol{y}_i, c_i)} \left(\frac{p_j(\boldsymbol{y}_i)^2}{\sigma_j^2} - 1 \right),
\end{aligned}$$

where $p_j(\boldsymbol{y}) = \boldsymbol{v}_j \cdot \boldsymbol{y} - r_j$, $\boldsymbol{v}_j = (\cos(\phi_j), \sin(\phi_j))$, and $\boldsymbol{w}_j = (-\sin(\phi_j), \cos(\phi_j))$. Formulas for computing J_j and its partial derivatives are given in Appendix. As before, we use the constrained optimization function *fmincon* in MATLAB with multiple initializations to search for the best lines. After estimating $\boldsymbol{\theta}$, we can estimate the probability that any observed data pair $(\boldsymbol{y}_i, c_i)$ corresponds to a target by

$$1 - \frac{\lambda f_0(\boldsymbol{y}_i) g_0(c_i)}{\xi(\boldsymbol{y}_i, c_i)}.$$

3.2. Examples

We demonstrate the above gradient-based, constrained optimization of H using the same dataset as in the previous section, except here we use additionally the ATR scores associated with the given points. Once again, we refer to Section 4 for a detailed description of the data. With respect to the extensions provided in the previous subsection (Section 3.1), the model configuration is $f_0 = 1, g_0$ and g_1 non-constant, $f_1 = 0$, and $k = 1$. The results are shown in Fig. 3. Here, the bottom row is analogous to the three panels of Fig. 2, and the top row consists of some extra images to visualize the ATR score data. The top-left panel shows the two ATR score density curves, where the clutter score density $g_0(c)$ is given in blue and the target score density $g_1(c)$ is given in red. These densities were estimated using a kernel estimate from an independent observation. The center panel shows the data points colorized according to their ATR score, with the colorbar on the right side of the panel. The right panel shows the data points colorized according to their ATR score log-likelihood ratio $\log(g_1(c_i)/g_0(c_i))$, where red circles represent those points with this value greater than zero and the blue stars represent those with this value less than zero. In other words, the red points have an ATR score that is more target-like than clutter-like, and the blue points have an ATR score that is more clutter-like than target-like. One can see that for this particular dataset, the two densities are fairly similar, and therefore, the separability of the data based on ATR score alone is quite difficult. Since the points with target-like ATR score are spread throughout the entire area instead of being concentrated along one line, the best fitted line is one that splits the area roughly in half and has a large value of σ. The three model parameters not associated with the line are now estimated to be $\hat{\lambda} = 337.1$, $\hat{\gamma} = 299.1$, and $\hat{\sigma} = 0.3220$.

In situations when the clutter points follow a Poisson process but not with a constant intensity, fitting the model with a constant f_0 can lead to undesirable results. Often times the best fit line is forced to run through a region of high density clutter points. By extending the model to include inhomogeneous clutter (non-constant f_0), any high density clutter regions

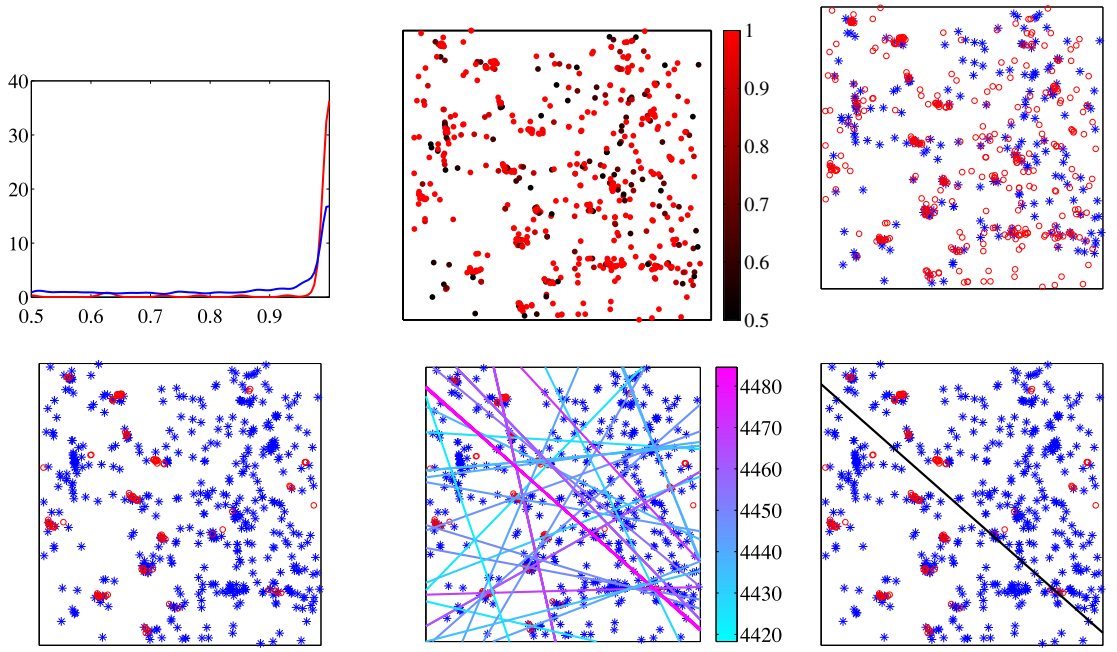

Fig. 3. Estimation of target-layer trajectory using ATR scores. The top-left panel shows the ATR score densities $g_0(c)$ (blue) and $g_1(c)$ (red) for this dataset. The top-center panel shows the data colorized according to ATR score. The top-right panel shows the data colorized according to $\text{sign}(\log(g_1(c_i)/g_0(c_i)))$. The bottom row is analogous to Fig. 2. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

are considered background, and the MLE is allowed to pick out any truly linear pattern that stands out from the clutter. Fig. 4 illustrates the difference in results obtained from using homogeneous versus inhomogeneous f_0 on three simulated datasets with inhomogeneous clutter and a simulated target line. In the first two datasets the full point cloud is simulated from the model, but in the third dataset we use real clutter and insert a simulated target line. The left column shows an image of $f_0(\mathbf{y})$ for each dataset, the center column shows the best fitted line using the homogeneous clutter model, and the right column shows the best fitted line using the model with the inhomogeneous f_0 shown on the left. In all cases the true target line was missed in the homogeneous case, and the fitted line was drawn instead through the areas of high density clutter. The problem is rectified in the inhomogeneous case, and in all cases the gradient descent finds the correct target line.

After examining the difference in results obtained from using the homogeneous versus inhomogeneous clutter model, we revisit the example shown in Fig. 3. In Fig. 5 we run the optimization with inhomogeneous f_0 instead (keeping all other model configurations the same) and obtain more reasonable results. The function f_0 is estimated via a kernel estimate from the whole data, using the isotropic Gaussian kernel and with bandwidth large enough to minimize the effects of any line present in the data. (This same procedure is used to estimate f_0 in all the cases with real data and non-constant f_0.) Also, we maintain that $\int_U f_0(u)du = |U|$, which in this example is equal to 1. The gradient descent now picks out the most prominent target line in the scene. Compare the center and right panels of Fig. 5 with the last two panels of Fig. 3. In this case, $\hat{\lambda} = 542.34$, $\hat{\gamma} = 65.23$, and $\hat{\sigma} = 0.0113$, which shows that the MLE includes far less target points along the line relative to the case with constant f_0.

Finally, in Fig. 6 we show an example of fitting the extended model with more than one target line. Here, we use fully simulated data from the model with homogeneous f_0, g_0 equal to the standard normal distribution, g_1 equal to the normal distribution with mean 1 and variance 1, $f_1 = 0$, and $k = 2$ target lines. Clearly, the optimization procedure finds the correct two target lines in this case.

4. Numerical results

In this section we present some experimental results on target-layer trajectory estimation using both point locations and ATR scores. We will use a number of datasets to evaluate the algorithm. Since in the case of real data, it is difficult to quantify the performance due to the lack of a ground truth target line, we also rely on simulated data to evaluate the estimation procedure. We will use three different scenarios: (1) both target and clutter points, and their associated ATR scores, are simulated from their respective models, (2) the target locations and their associated ATR scores are simulated while the clutter data is taken from a real dataset, and (3) both the target and clutter points, and their associated ATR scores, are taken from a real dataset.

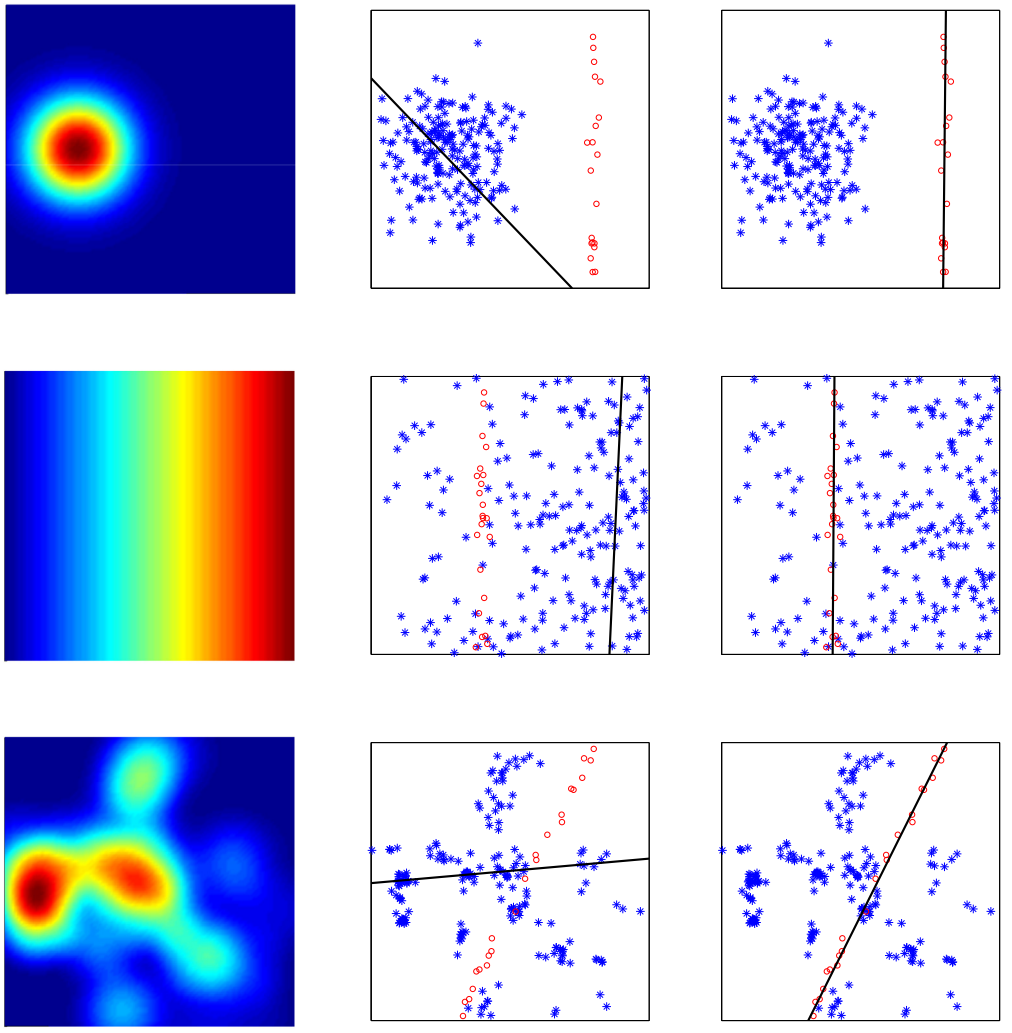

Fig. 4. Three examples of model fitting using non-constant clutter intensity f_0. Left: Plot of f_0. Center: Best fitted line using model with homogeneous background clutter. Right: Best fitted line using model with the inhomogeneous f_0 shown at left.

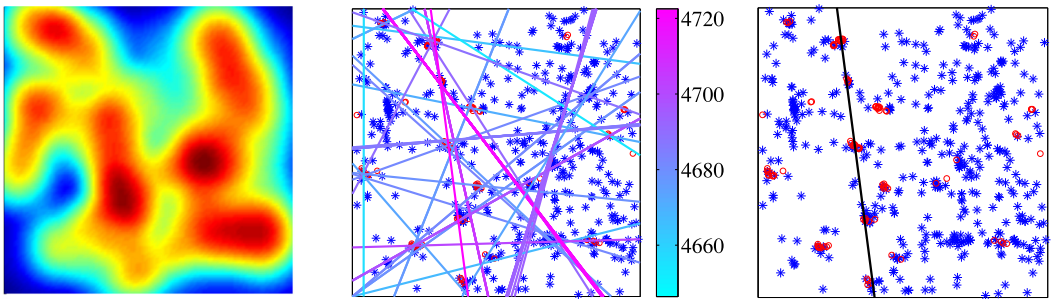

Fig. 5. Estimation of target-layer trajectory using ATR scores and non-constant f_0. Left: Plot of f_0. Center: Local solutions resulting from different initializations of the line, colorized according to the H value. Right: The best solution amongst the local results. Compare this result to the bottom row of Fig. 3, which shows results from using a constant f_0. (Note that the region U is centered at the origin in this example.) (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4.1. Simulated targets with simulated clutter data

In this experiment, clutter points and target points are simulated according to the line trajectory model with ATR scores described in Section 3. In Fig. 7, one can see the two simulated datasets in the left column. The clutter points are simulated

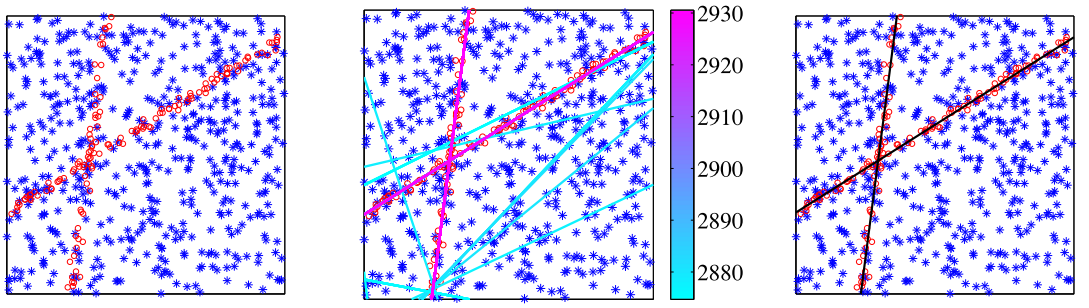

Fig. 6. Estimation of two target-layer trajectories amidst homogeneous clutter. Left: Plot showing clutter points (blue) and target points (red). Center: Local solutions resulting from different pairs of initializations, colorized according to the H value. Right: The best solution pair amongst the local results. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

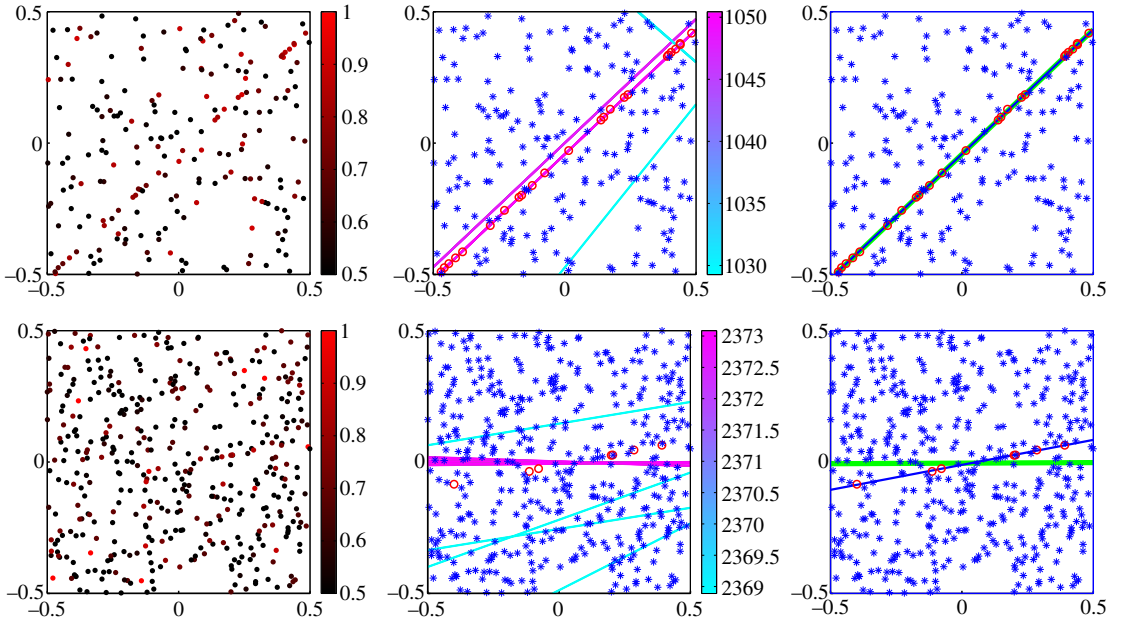

Fig. 7. Estimation of target-layer trajectory using ATR scores in fully simulated data. Each row represents results from a different dataset. Left: the original points colorized according to ATR score; center: estimated lines from multiple initializations, right: the original ground truth line in blue with best estimated line in black. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

from a homogeneous Poisson point process. There are expected to be 50 clutter points in the top row of Fig. 7, and 100 clutter points in the bottom row. The target points are simulated using a Poisson point process along a randomly generated line and then perturbing the points away from the line using a Gaussian distribution with $\sigma = 0.03$. The occurrence rates for target locations are selected so that targets are on average spaced 0.1080 units apart in the top row and 0.2700 units apart in the bottom row. The ATR scores are generated by resampling with replacement from a real dataset that contains ATR scores from both targets and clutter. Although the original ATR scores take values in [0, 1], the data points with ATR score of less than 0.5 are removed in the pre-processing, as in the previous cases.

In the left panels, one can see the points with their respective ATR scores. Black indicates that the ATR score is low, and red indicates that the ATR score is high. Note the relative difficulty, even for a human observer, in estimating where the line is located. In the middle column of Fig. 7, we show the 50 estimated lines (not all distinct) obtained using random initializations in each case. The color of the line indicates the value of the log-likelihood function H. The red circles denote the true target locations, and the blue stars denote the locations of clutter points. In the far right panels the true line used to simulate the data is plotted in blue, and the best fitted line is plotted in black.

4.2. Real clutter and simulated target locations

In this experiment we add simulated target locations to real clutter data in order to evaluate the performance of the line estimation procedure. The clutter data is obtained by taking a subset of data collected by a side-scan sonar system deployed by NSWC PCD that contains real ATR scores from detected contacts. Since target locations are known using manual detection

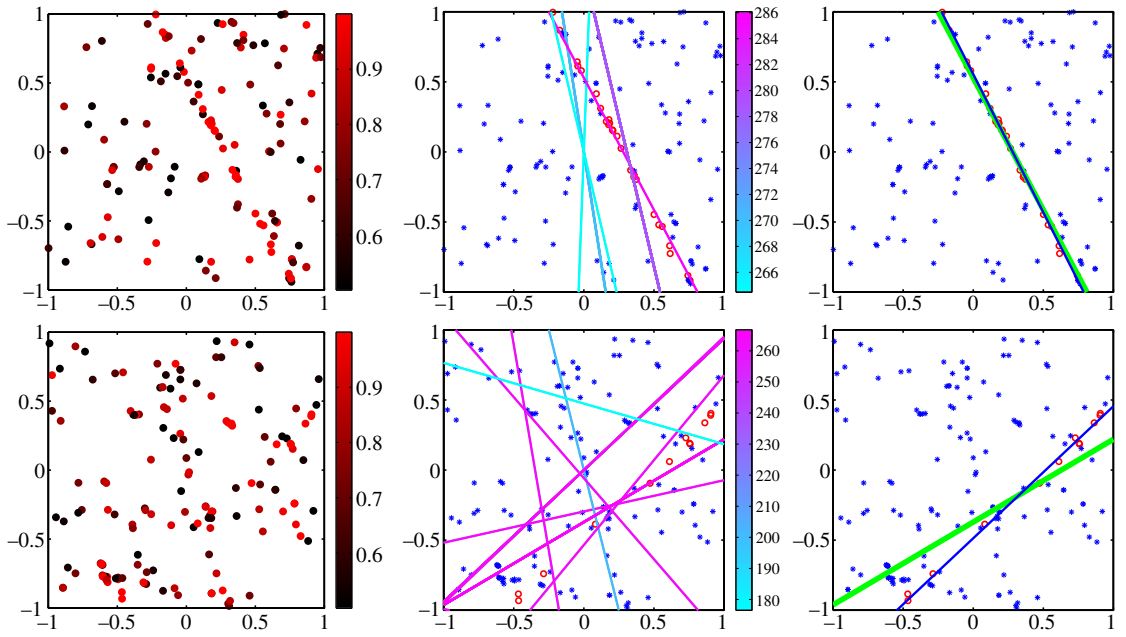

Fig. 8. Estimation of target-layer trajectory using ATR scores in data with real clutter and simulated targets. Each row represents results from a different dataset. Left: the original points colorized according to ATR score; center: estimated lines from multiple initializations, right: the original ground truth line in blue with best estimated line in black. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

in this data, we obtained a sample of clutter by simply removing points related to targets. Also, as a preprocessing step, points that have an ATR score of less than 0.5 are removed.

The simulated target locations are generated over the region by simulating from a Poisson point process with rate γ along the line and applying Gaussian noise with covariance $\sigma^2 I$ to the points on the line. The plots in Fig. 8 are laid out similarly to those in Fig. 7 and display results on two datasets. The ATR scores for the simulated target data are generated by sampling with replacement from a list of the ATR scores of true targets from the NSWC PCD dataset. The targets in the top row are generated with an average spacing of 0.15 units, and the bottom row has an average spacing of 0.25 units, where the region size is 2 units $\times 2$ units. These examples are challenging because the spacing between the targets is relatively large compared to the size of the region. For an increased target spacing, the occurrence of targets is relatively sparse and it makes trajectory estimation more difficult. Despite this, the algorithm performs reasonably well in the cases presented in Fig. 8.

As shown in light blue lines in the center panels of Fig. 8, the optimization is local and solutions far from the global solution are often selected. However, we can overcome this limitation using multiple random initializations, as earlier, and reach a global solution. To quantify the line estimation performance, there are several ideas. One is to compare the parameters of the estimated line with those of the ground truth. A better measure seems to be a metric that compares the two lines as sets of points. Towards that goal, the Hausdorff distance $d_H(\cdot, \cdot)$ computes the maximum orthogonal projection of one point in either line to the other line. For two line segments $L_1, L_2 \subset U$, this distance is given by

$$d_H(L_1, L_2) = \max\{\max_{x \in L_1} \min_{y \in L_2} \|x - y\|, \max_{y \in L_2} \min_{x \in L_1} \|x - y\|\}.$$

Since L_1 and L_2 are line segments, $d_H(L_1, L_2)$ will be maximized at the boundary of U if U is bounded and convex. The left and center panels of Fig. 9 show two visualizations of the Hausdorff distance calculation. The solid blue lines are L_1 and L_2, and the dashed lines represent maximal projection distances from one line to the other, contained within U. The red dashed line represents the maximum of these projection distances and thus represents the Hausdorff distance that is measured in both cases. Here, $U = [-1, 1] \times [-1, 1]$, and the Hausdorff distances measured are 1.34 and 0.96 in the left and center panels, respectively.

The right panel of Fig. 9 plots the Hausdorff distances measured from the following experiment. One thousand point clouds were simulated from the baseline model (without ATR scores) using randomly generated parameters ϕ, r, and γ, a fixed a set of clutter, and a fixed $\sigma = 0.05$. For each dataset, a best fit line is obtained from multiple initializations, and the result is compared to the true line using the above Hausdorff distance between the two lines. The obtained distances are then plotted against the number of simulated targets n. The bottom curve represents the first quartile, the middle represents the median, and the top represents the third quartile. One can see that, as the number of targets increases, the Hausdorff distance as well as the interquartile range both tend to decrease, indicating an increase in accuracy of our estimated line.

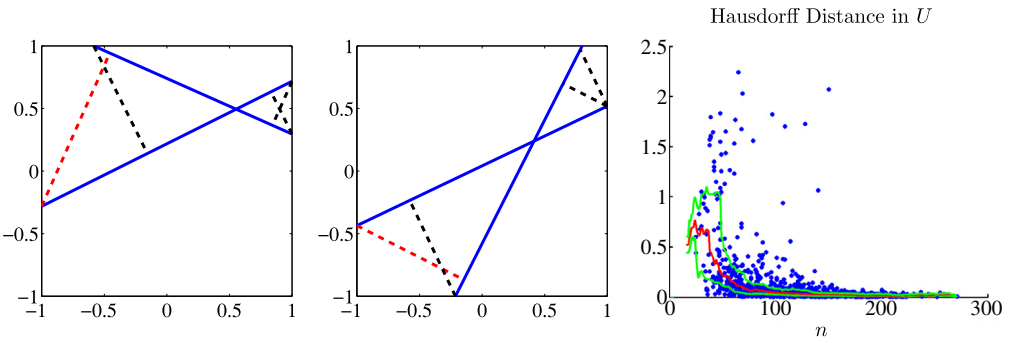

Fig. 9. Hausdorff distance between lines in U. The left and center panels show an example of the Hausdorff distance calculation. The right panel shows Hausdorff distances between true and estimated lines, for 1000 realizations, plotted against the number of targets n. The three curves represent the sliding quartiles using a window size of 20 with the median plotted in red. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4.3. Real targets and clutter data

Next, we test our estimation procedure on two datasets of contacts obtained from an ATR system developed by NSWC PCD. Acoustic data was collected from an autonomous underwater vehicle (AUV) equipped with a high resolution, high frequency, synthetic aperture sidescan sonar. The vehicle traveled in a uniformly-spaced search pattern to cover the entire test field, which was approximately one nautical square mile and contained about 15–20 targets of interest laying on the seafloor in each case. The raw sonar data was then post-processed to form a complex-valued image via a k-space or wave number beamformer (see Ch. 6 of Soumekh, 1999). The sonar imagery was then fed to an onboard ATR algorithm that detected and output all potential target locations, or contacts. Associated with each contact, the algorithm also output a classification score from 0 to 1 indicating the likelihood of it being a target of interest. We direct the reader to the Refs. Tucker and Azimi-Sadjadi (2011) and Isaacs and Tucker (2011) for more information on the detection and classification procedures within the ATR algorithm. The data in Figs. 2, 3, 10, 11, and 13 represent the contact locations from these datasets that have a score greater than 0.5, i.e., those contacts with a positive target classification. Datasets 1 and 2 differ in the following ways: they were collected in different locations, the target fields were different, and a different sonar system was used in each case. All of these factors contribute to a greater occurrence of background clutter in dataset 2 compared to that of dataset 1, thus making for an interesting comparison of results from fitting our model.

1. **Baseline model without and with ATR scores**: Figs. 10 and 11 show the results of our estimation under the baseline model on real dataset 1, and they are analogous to Figs. 2 and 3, which show results on real dataset 2. For real dataset 1 we have $m = 471$ and $U = [0, 1] \times [0, 1]$, and without using ATR scores the three parameters are estimated to be $\hat{\lambda} = 347.1$, $\hat{\gamma} = 96.99$, and $\hat{\sigma} = 0.006572$. When the ATR scores are included in the analysis, we get $\hat{\lambda} = 196.2$, $\hat{\gamma} = 279.7$, and $\hat{\sigma} = 0.2140$. Similar to the results for real dataset 2 shown in Section 3.2, since there are many points with target-like ATR scores scattered widely throughout U and not necessarily in a compact linear fashion, the value of $\hat{\sigma}$ is much larger when ATR scores are considered compared to when they are not.
2. **Extended model using inhomogeneous clutter**: However, we observe (in Fig. 12) a similar phenomenon as in Section 3.2 when we include inhomogeneous f_0 in the model, whereby the MLE assigns more points as background clutter and less as members of a target line when compared to the MLE obtained with homogeneous f_0. Fig. 12 is analogous to Fig. 5, and likewise one can compare Fig. 12 with the last two panels of Fig. 11. In this case $\hat{\lambda} = 366.6$, $\hat{\gamma} = 81.72$, and $\hat{\sigma} = 0.006270$, which is similar to the results obtained with homogeneous f_0 and without ATR scores shown in Fig. 10 but with a slightly more selective target line.
3. **Extended model using background target process**: It is often possible in real data to have target locations that are not associated with the trajectory being estimated, and this appears to be the case with real datasets 1 and 2. Thus, a more appropriate model would be one that includes the extension detailed in Section 3.1 of a non-zero f_1, i.e., the inclusion of a background target process. As the inclusion of an inhomogeneous f_0 rectifies the poor results shown in Figs. 3 and 11, so does the inclusion of a homogeneous f_1. Here, we compute the MLE of $\boldsymbol{\theta}$ for real datasets 1 and 2 using the model with ATR scores and homogeneous f_0 and f_1. Fig. 13 shows the best fit line in each case, and they are nearly identical to the best fit lines from Figs. 5 and 12. Tables 1 and 2 summarize the results for both real datasets under various model configurations presented in this paper.

5. Comparison with RANSAC

Random sample consensus (RANSAC) (Fischler and Bolles, 1981) is an approach for fitting geometric models to noisy point cloud data, and is widely used in computer vision applications to find lines and curves. Unlike our proposed line

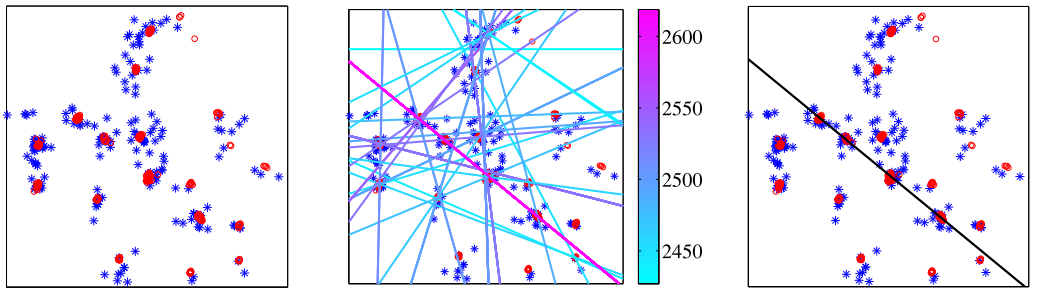

Fig. 10. Estimation of target-layer trajectory on real dataset 1 without using ATR scores. This figure is analogous to Fig. 2 in Section 2.

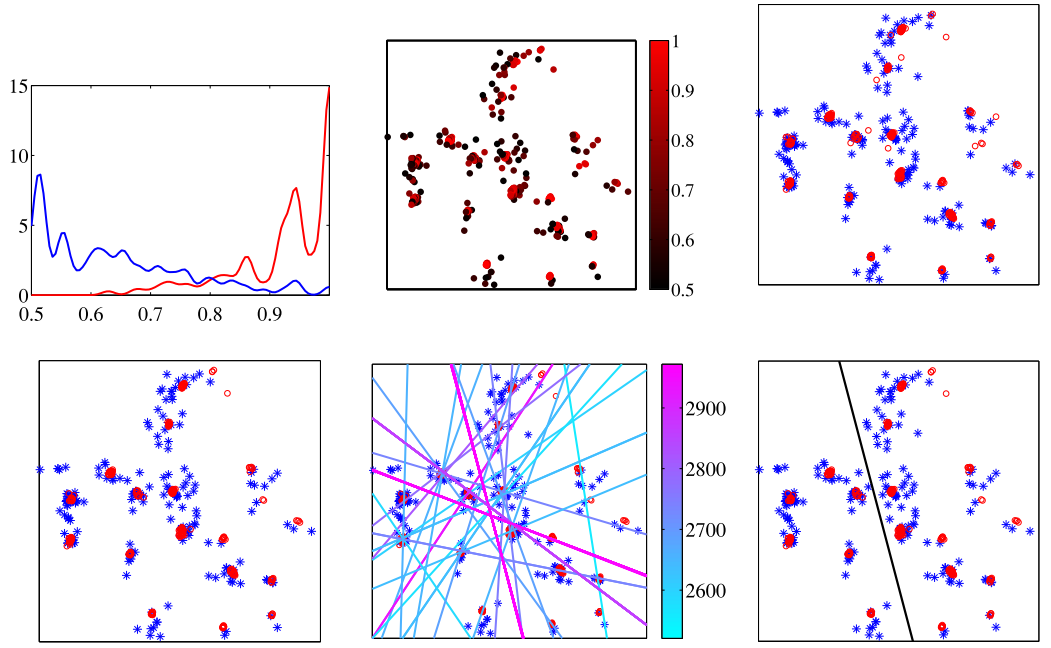

Fig. 11. Estimation of target-layer trajectory on real dataset 1 using ATR scores and assuming homogeneous clutter. This figure is analogous to Fig. 3 in Section 3.

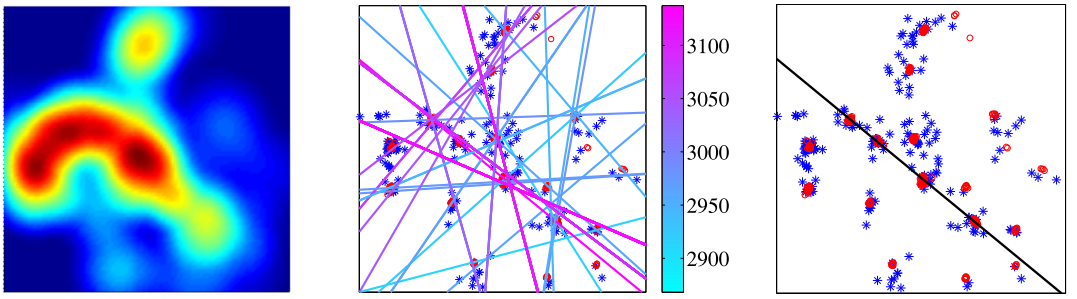

Fig. 12. Estimation of target-layer trajectory on real dataset 1 using ATR scores and inhomogeneous clutter model (non-constant f_0). This figure is analogous to Fig. 5 in Section 3. Compare to the bottom row of Fig. 11, which shows results from using a constant f_0.

model, the standard RANSAC algorithm does not handle marks such as ATR scores in the estimation, and thus we make a simple modification to standard RANSAC to incorporate this additional information when applicable. Let c_i denote the ATR score, and let $w_i = c_i / \sum_{i=1}^{m} c_i$ denote the weight of the ith data point based on its ATR score. Instead of uniformly sampling the data points, a core step in standard RANSAC, we perform a weighted sampling of the points so that the ith point has probability w_i of being selected. Furthermore, instead of just counting the number of elements in the consensus set $\tilde{S}$, we compute a weighted measure of the set as $\sum_{i \in \tilde{S}} w_i$. We will call this modified algorithm Weighted RANSAC, or WRANSAC, and will compare results from our model-based approach to results from standard RANSAC and WRANSAC.

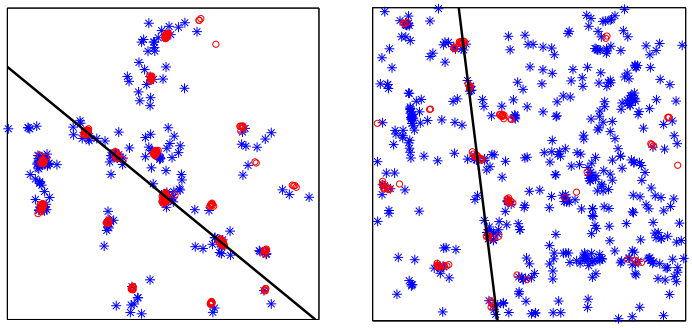

Fig. 13. Estimation of target-layer trajectory using ATR scores and homogeneous background target process. Left: best line estimated from real dataset 1. Right: best line estimated from real dataset 2.

Table 1
Results of model parameter estimation on real datasets 1 and 2 using homogeneous f_0. We show results from the cases without ATR scores, with ATR scores, and with ATR scores and homogeneous background target process f_1.

Dataset	Parameter	Baseline w/o ATR	Baseline w/ATR	Extension w/ATR & bkgd targets
	$\hat{\lambda}$	347.1	196.2	196.8
	$\hat{\gamma}$	96.99	279.7	82.14
# 1	$\hat{r}$	0.6269	0.5328	0.6270
($m = 471$)	$\hat{\phi}$	0.8823	0.2580	0.8838
	$\hat{\sigma}$	0.006572	0.2140	0.005776
	$\hat{\beta}$	N/A	N/A	171.2
	$\hat{\lambda}$	551.3	337.1	334.9
	$\hat{\gamma}$	56.26	299.1	54.91
# 2	$\hat{r}$	0.3984	0.6948	0.3963
($m = 608$)	$\hat{\phi}$	0.1264	0.8472	0.1239
	$\hat{\sigma}$	0.008241	0.3220	0.008626
	$\hat{\beta}$	N/A	N/A	217.8

Table 2
Results of model parameter estimation on real datasets 1 and 2 using inhomogeneous f_0. We show results from the cases without ATR scores, with ATR scores, and with ATR scores and homogeneous background target process f_1.

Dataset	Parameter	Baseline w/o ATR	Baseline w/ATR	Extension w/ATR & bkgd targets
	$\hat{\lambda}$	377.7	366.6	220.5
	$\hat{\gamma}$	72.94	81.72	78.13
# 1	$\hat{r}$	0.6275	0.6267	0.6270
($m = 471$)	$\hat{\phi}$	0.8825	0.8821	0.8842
	$\hat{\sigma}$	0.005212	0.006270	0.005541
	$\hat{\beta}$	N/A	N/A	150.6
	$\hat{\lambda}$	558.2	542.3	261.8
	$\hat{\gamma}$	49.37	65.23	55.08
# 2	$\hat{r}$	0.3957	0.3989	0.3957
($m = 608$)	$\hat{\phi}$	0.1227	0.1296	0.1230
	$\hat{\sigma}$	0.007407	0.01127	0.008361
	$\hat{\beta}$	N/A	N/A	290.7

We perform 100 independent trials for each combination of $\sigma \in \{5.4 \times 10^{-3}, 1.1 \times 10^{-1}\}$, $\lambda \in \{10, 100\}$, and $\gamma \in \{10, 50\}$, and $U = [0, 1] \times [0, 1]$, for a total of 800 independent trials. In each trial we simulate dataset according to the chosen parameter values under our model and fit both RANSAC and WRANSAC as well as our baseline model. In these datasets the underlying linear trajectories were generated randomly, and the densities g_0, g_1 were the same as those shown in Fig. 3. We then evaluate the estimation performance in terms of Hausdorff distance of the estimated line from the ground truth line. The top panel of Fig. 14 shows an overall summary of the results in the form of a boxplot. Clearly, on average our model outperforms both RANSAC and WRANSAC in this experiment. In fact, our model outperforms RANSAC and WRANSAC under all eight simulation scenarios.

We perform a similar experiment using real background clutter and simulated target points. An example that compares the WRANSAC estimate with our model-based solution involving inhomogeneous clutter is shown in Fig. 15. As one can see, the estimate from our proposed method is closer to the ground truth line than that of WRANSAC. Since the background clutter comes from real data, there is no guarantee that the clutter is distributed homogeneously, and thus we make use

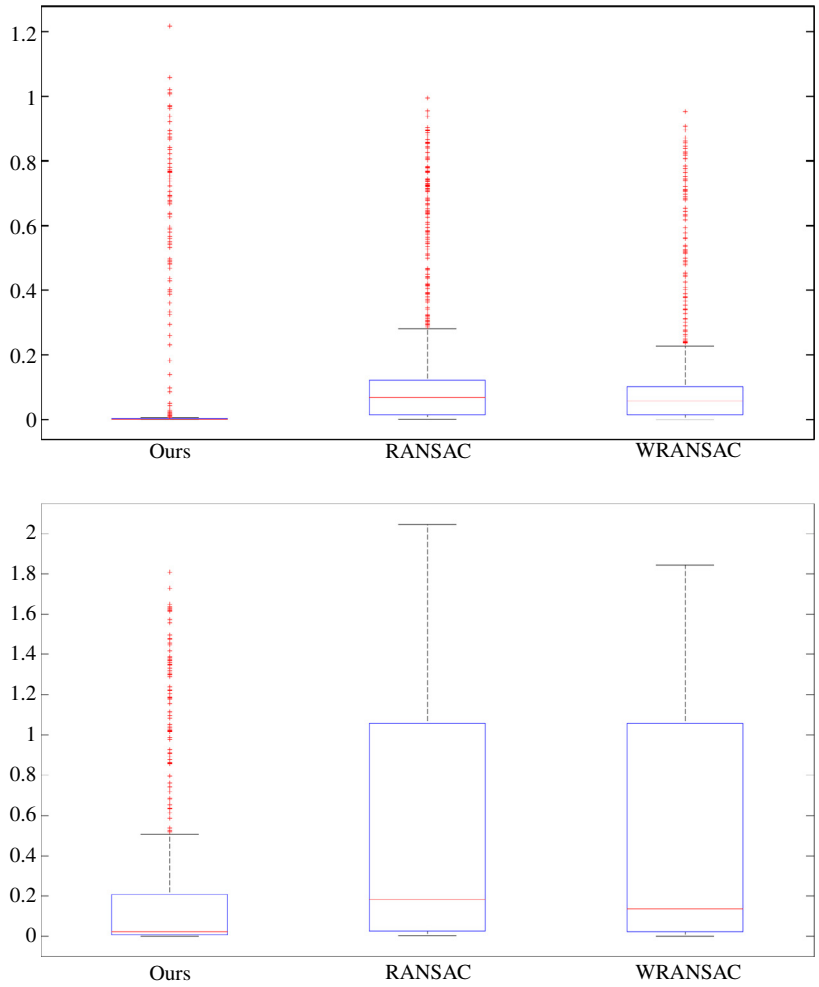

Fig. 14. Boxplots of Hausdorff distances from ground truth to estimated lines obtained from fitting our line model, RANSAC, and WRANSAC. The top panel shows the summary of 800 trials from fully simulated data with homogeneous background clutter. The bottom panel shows the summary from 400 independent trials using data that contains real background clutter and simulated trajectory points.

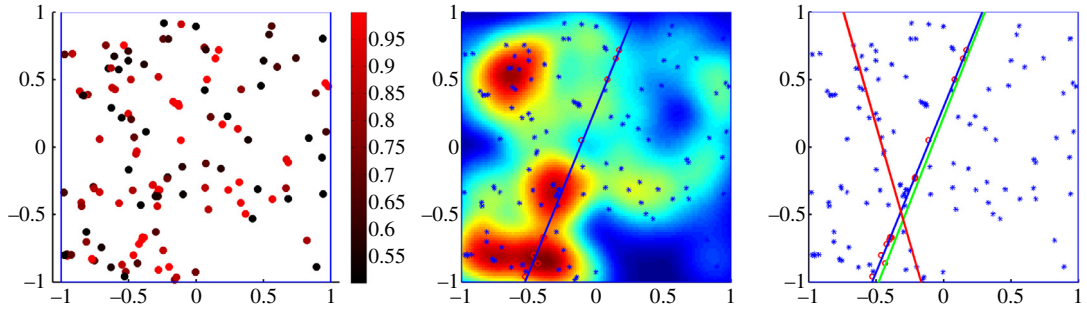

Fig. 15. Comparison of the line fit obtained from our model and from WRANSAC on real simulated data. The left panel shows the data with real background clutter and simulated targets along a line, weighted according to ATR score. The middle panel shows the true simulated line and the background clutter map f_0. The right panel shows the true line in blue, the WRANSAC line in red, and our proposed method's line in green. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

of our model extension with an inhomogeneous background clutter model f_0. To measure the overall performance, we randomly generated 100 lines for each combination of parameter values $\sigma \in \{0.005, 0.1\}$ and $\gamma \in \{10, 50\}$ over randomly selected regions of real observed background clutter. Note that λ is not applicable to this experiment since the background clutter comes from real data. The bottom panel of Fig. 14 summarizes the results of this experiment, and as before the model-based methods outperform both RANSAC and WRANSAC.

6. Hypothesis test for line detection

Here, we design a procedure to test the null hypothesis that our data does not contain a target-layer line versus the alternative that it does. Following the notation in Section 2 and given data $(\mathbf{Y}, \mathbf{c}) = \{(\mathbf{y}_1, c_1), \ldots, (\mathbf{y}_m, c_m)\}$, the likelihood function under the null model is noted by $Q((\mathbf{Y}, \mathbf{c})|\boldsymbol{\theta_0})$, and the likelihood function under the proposed line model is noted by $P((\mathbf{Y}, \mathbf{c})|\boldsymbol{\theta_0})$. The parameters included in the vector $\boldsymbol{\theta}$ and the formula for P depend on the model extensions one wishes to include from those presented in Section 3.1. For example, if the model includes the background target process but only has one line, the parameter vector is given as $\boldsymbol{\theta} = (\lambda, \gamma, r, \phi, \sigma, \beta)$, and the likelihood function is given as

$$P((\mathbf{Y}, \mathbf{c})|\boldsymbol{\theta}) = e^{-\lambda A - \beta B - \gamma J} \prod_{i=1}^{m} (\lambda f_0(\mathbf{y}_i) g_0(c_i) + \beta f_1(\mathbf{y}_i) g_1(c_i) + \gamma \alpha_\sigma(\mathbf{y}_i) g_1(c_i)).$$

We design the null model to consist of terms in the proposed model that do not relate to the line. Therefore, if the proposed model does not include the background target process, then $\boldsymbol{\theta_0} = \lambda$, and the likelihood for the null model is given as

$$Q_1((\mathbf{Y}, \mathbf{c})|\boldsymbol{\theta_0}) = e^{-\lambda A} \prod_{i=1}^{m} (\lambda f_0(\mathbf{y}_i) g_0(c_i)).$$

If the proposed model contains the background target process, then $\boldsymbol{\theta_0} = (\lambda, \beta)$, and the likelihood for the null model is given as

$$Q_2((\mathbf{Y}, \mathbf{c})|\boldsymbol{\theta_0}) = e^{-\lambda A - \beta B} \prod_{i=1}^{m} (\lambda f_0(\mathbf{y}_i) g_0(c_i) + \beta f_1(\mathbf{y}_i) g_1(c_i)).$$

One can perform the hypothesis test of a proposed line model that includes the background target process versus null model 1; however, the tested hypothesis would then be that of whether there are targets present in the scene or not. The hypothesis test is no longer specifically testing for a target line in this case.

As explained in Section 2, the usual regularity conditions that ensure validity of the asymptotic chi-squared distribution of $D = -2\log(\text{GLRT})$, where GLRT is given in Eq. (1), do not apply in our situation. Therefore, we rely on a Monte Carlo simulation to compare the value of $D = d^*$ (the value of D associated with the data) to an empirical distribution generated under the null assumptions. The procedure is as follows:

1. Given the data $(\mathbf{Y}, \mathbf{c}) = \{(\mathbf{y}_1, c_1), \ldots, (\mathbf{y}_m, c_m)\} \subset U \times \mathbb{R}_+$, null model $i = 1$ or 2, clutter distribution $f_0(\mathbf{y})$, background target distribution $f_1(\mathbf{y})$, clutter ATR density $g_0(c)$, and target ATR density $g_1(c)$, compute the GLRT statistic
$$d^* = -2\log\left(\frac{\max_{\boldsymbol{\theta_0}} Q_i((\mathbf{Y}, \mathbf{c})|\boldsymbol{\theta_0})}{\max_{\boldsymbol{\theta}} P((\mathbf{Y}, \mathbf{c})|\boldsymbol{\theta})}\right).$$
2. Generate $\{d_j, j = 1, \ldots, N\}$, a set of N random samples from the distribution of D under the null hypothesis, via the following.
   (a) If $i = 1$ (null model 1), for $j = 1, \ldots, N$,
      i. Simulate a dataset with m points from the null model. That is, generate m clutter points $(\mathbf{Z}_j, \tilde{\mathbf{c}}_j) = \{(\mathbf{z}_{j,1}, \tilde{c}_{j,1}), \ldots, (\mathbf{z}_{j,m}, \tilde{c}_{j,m})\}$, with $\mathbf{z}_{j,k}$ i.i.d. from f_0 and $\tilde{c}_{j,k}$ i.i.d. from g_0.
      ii. Compute the GLRT statistic
      $$d_j = -2\log\left(\frac{\max_{\boldsymbol{\theta_0}} Q_1((\mathbf{Z}_j, \tilde{\mathbf{c}}_j)|\boldsymbol{\theta_0})}{\max_{\boldsymbol{\theta}} P((\mathbf{Z}_j, \tilde{\mathbf{c}}_j)|\boldsymbol{\theta})}\right).$$
   (b) Else if $i = 2$ (null model 2), for $j = 1, \ldots, N$,
      i. Set the number of clutter points $m_{0,j}$ to be a Binomial(m, p) random variable with $p = \hat{\lambda}A/(\hat{\lambda}A + \hat{\beta}B)$ where $(\hat{\lambda}, \hat{\beta}) = \text{argmax}_{\lambda,\beta} Q_2((\mathbf{Y}, \mathbf{c})|\lambda, \beta)$, and set the number of background mines to be $m_{1,j} = m - m_{0,j}$.
      ii. Simulate a dataset with m points from the null model via the following. Generate $m_{0,j}$ clutter points $(\mathbf{z}_{j,1}, \tilde{c}_{j,1}), \ldots, (\mathbf{z}_{j,m_{0,j}}, \tilde{c}_{j,m_{0,j}})$ with $\mathbf{z}_{j,k}$ i.i.d. from the density on U proportional to f_0 and $\tilde{c}_{j,k}$ i.i.d. from g_0. Generate $m_{1,j}$ background target points $(\mathbf{z}_{j,m_{0,j}+1}, \tilde{c}_{j,m_{0,j}+1}), \ldots, (\mathbf{z}_{j,m}, \tilde{c}_{j,m})$ with $\mathbf{z}_{j,k}$ i.i.d. from the density on U proportional to f_1 and $\tilde{c}_{j,k}$ i.i.d. from g_1. Set $(\mathbf{Z}_j, \tilde{\mathbf{c}}_j)$ to be the union of the clutter and background target points.
      iii. Compute the GLRT statistic
      $$d_j = -2\log\left(\frac{\max_{\boldsymbol{\theta_0}} Q_2((\mathbf{Z}_j, \tilde{\mathbf{c}}_j)|\boldsymbol{\theta_0})}{\max_{\boldsymbol{\theta}} P((\mathbf{Z}_j, \tilde{\mathbf{c}}_j)|\boldsymbol{\theta})}\right).$$
3. Compute the empirical p-value as $p = |\{j : d_j > d^*\}|/N$, where $|\cdot|$ denotes cardinality.
4. Select a significance level α for the hypothesis test. If $p < \alpha$, then reject the null hypothesis and conclude that the data arise from the line model.

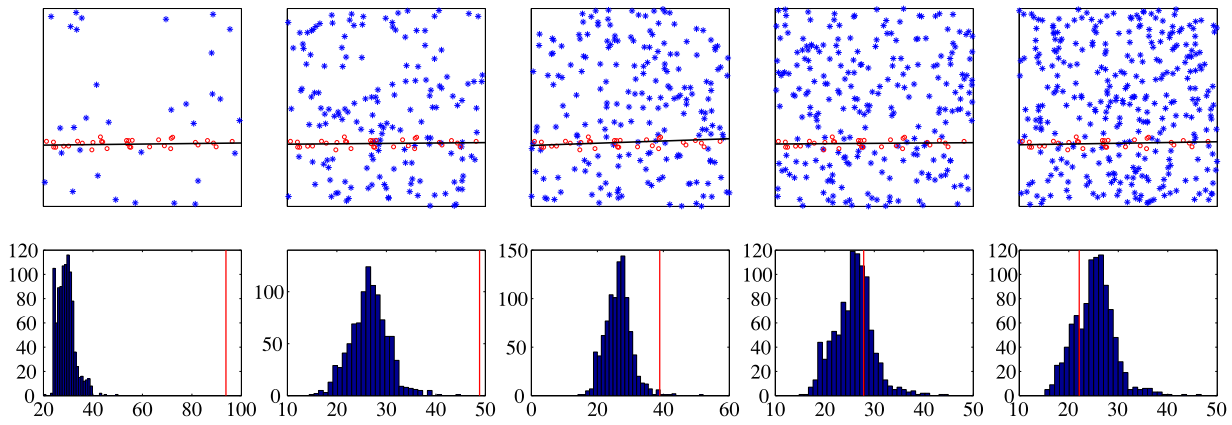

Fig. 16. Hypothesis testing on simulated data with increasing clutter level. Top row: best fit line for each case. Bottom row: histogram of samples from null distribution of D along with the value of d^* (vertical red line) for each respective case in the top row. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Remark. When computing the GLRT statistic in steps (1) and (2), the numerator is computed analytically in the case of null model 1 and via constrained numerical optimization with $\lambda, \beta \geq 0$ in the case of null model 2. In all instances the denominator is computed via the numerical optimization technique covered in Sections 2 and 3 of selecting the best (highest likelihood) solution from a fixed number of random initializations, n_{init}.

We carry out the hypothesis test outlined above on a series of simulated datasets. We generate the datasets by first randomly selecting the parameters γ, σ, r, and ϕ and simulating a target line in U. Then, we fix the target line and build each dataset by adding to the target line a random, uniformly generated set of clutter points. We forgo using ATR scores in the analysis here and thus set $g_0 = g_1 = 1$ in the likelihood formulas above. Since there are no ATR scores in the analysis, we do not consider the background target process in the proposed line model, and therefore we use null model 1 in the hypothesis testing. For this demonstration, we create five datasets with the same target line and an increasing number of clutter points, corresponding to a decreasing chance that the alternative hypothesis will be selected over the null in our hypothesis test.

The parameter values used to simulate the datasets are the following: $\gamma = 36, \sigma = 0.0161, r = 0.3179$, and $\phi = 1.5885$ on $U = [0, 1] \times [0, 1]$. The simulated target line yielded 32 points, and we set the increasing number of clutter points for each dataset to be 32, 160, 200, 240, and 320 for a total number of points $m = 64$, 192, 232, 272, and 352, respectively. For our hypothesis test procedure above, we set $n_{init} = 100$ and $N = 1000$. The top row of Fig. 16 shows the best fitted line overlayed on each dataset, and the bottom row shows the corresponding histogram of the d_j's generated from step (2) above along with the value of d^* indicated by the vertical red line. In each case the program fits the correct line to the data; however, the empirical p-values increase as the number of clutter points increases. The five empirical p-values are 0, 0, 0.006, 0.281, and 0.728, respectively.

Now, we investigate the effect of the selection of the clutter model f_0 on the hypothesis test results. Fig. 17 shows the results of model fitting and hypothesis testing on the three inhomogeneous clutter datasets shown in Fig. 4 with each target line removed. The top half of Fig. 17 shows results when we assume f_0 is homogeneous, and the bottom half shows results when f_0 is set to the appropriate inhomogeneous clutter distribution. Since the clutter is inhomogeneous by nature, the MLE when f_0 is set to the value 1 is a line that runs through an area of high density clutter, and the hypothesis test in all three cases yields a significant p-value. In other words, the hypothesis test yields a false positive in all three cases. This issue is resolved in the bottom half of Fig. 17 when using the correct inhomogeneous clutter model. All three hypothesis tests yield an empirical p-value greater than the standard significance level of $\alpha = 0.05$, and the null model correctly cannot be rejected in each case.

Additionally, we carry out the hypothesis test in a variety of model configurations on the two real datasets presented in Section 4.3, again with $n_{init} = 100$ and $N = 1000$. For each dataset we perform the hypothesis test procedure outlined above under the following four model scenarios, each with homogeneous f_0 and inhomogeneous f_0, for a total of eight tests.

1. Without ATR scores. Null model 1.
2. With ATR scores. Null model 1.
3. With ATR scores and homogeneous background targets. Null model 1.
4. With ATR scores and homogeneous background targets. Null model 2.

The plots of the histograms with GLRT statistic are shown for each case in Fig. 18, where "# 1" indicates real dataset 1, and "# 2" indicates real dataset 2. The empirical p-values are equal to 0 in all cases. A larger value of N is required to increase the numerical precision of the p-value calculation; however, a cost of increased computation time follows. Since hypothesis

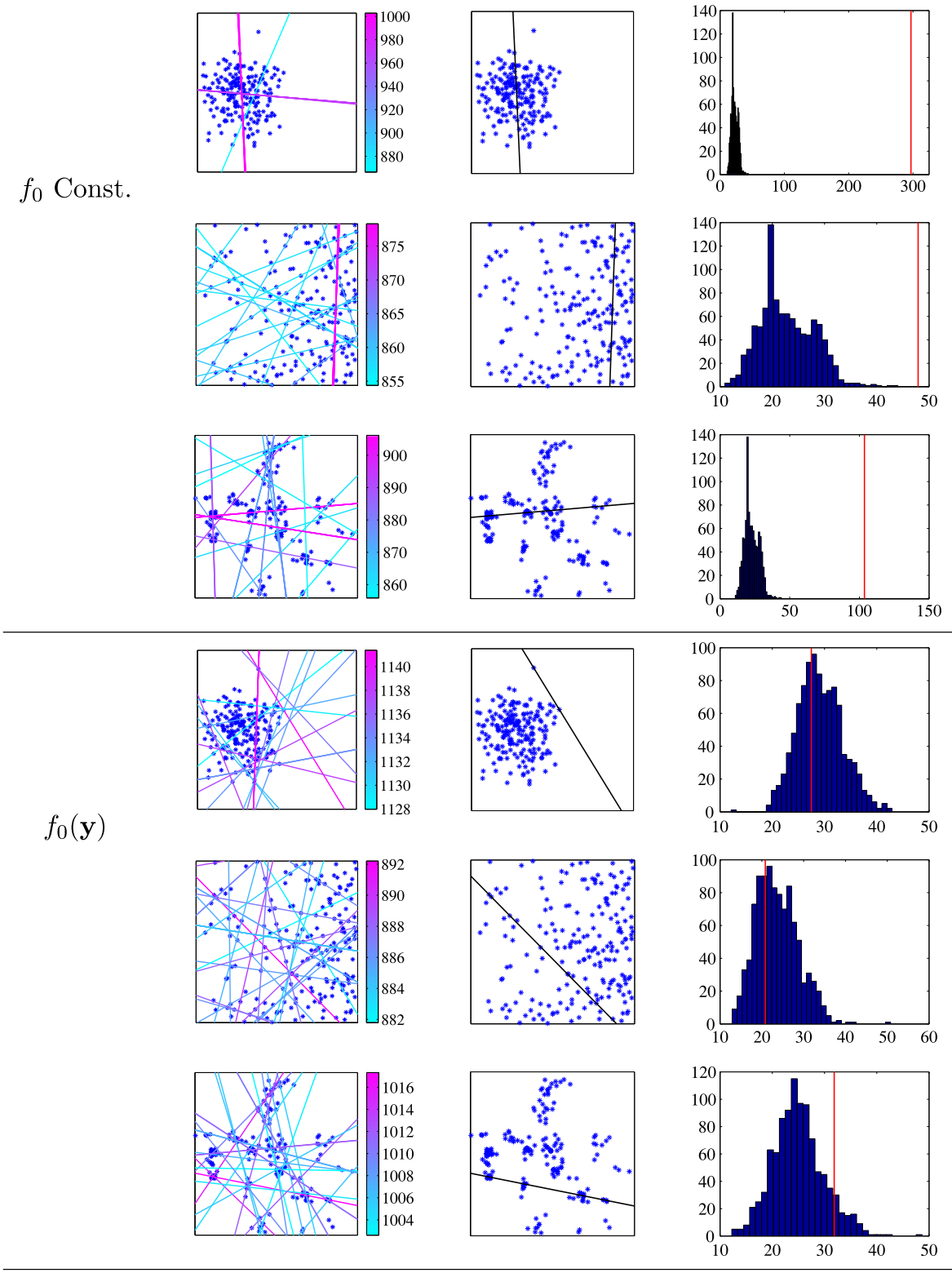

Fig. 17. Line fitting and hypothesis testing on inhomogeneous clutter data. Left: Optimization results from multiple initializations, colorized according to H value. Center: The best of the multiple solutions shown at left. Right: Histogram of samples from the null distribution of D along with the value of the test statistic d^* (vertical red line). Top half: Results with homogeneous clutter model. Bottom half: Results on the same data as the top half but with inhomogeneous clutter model. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

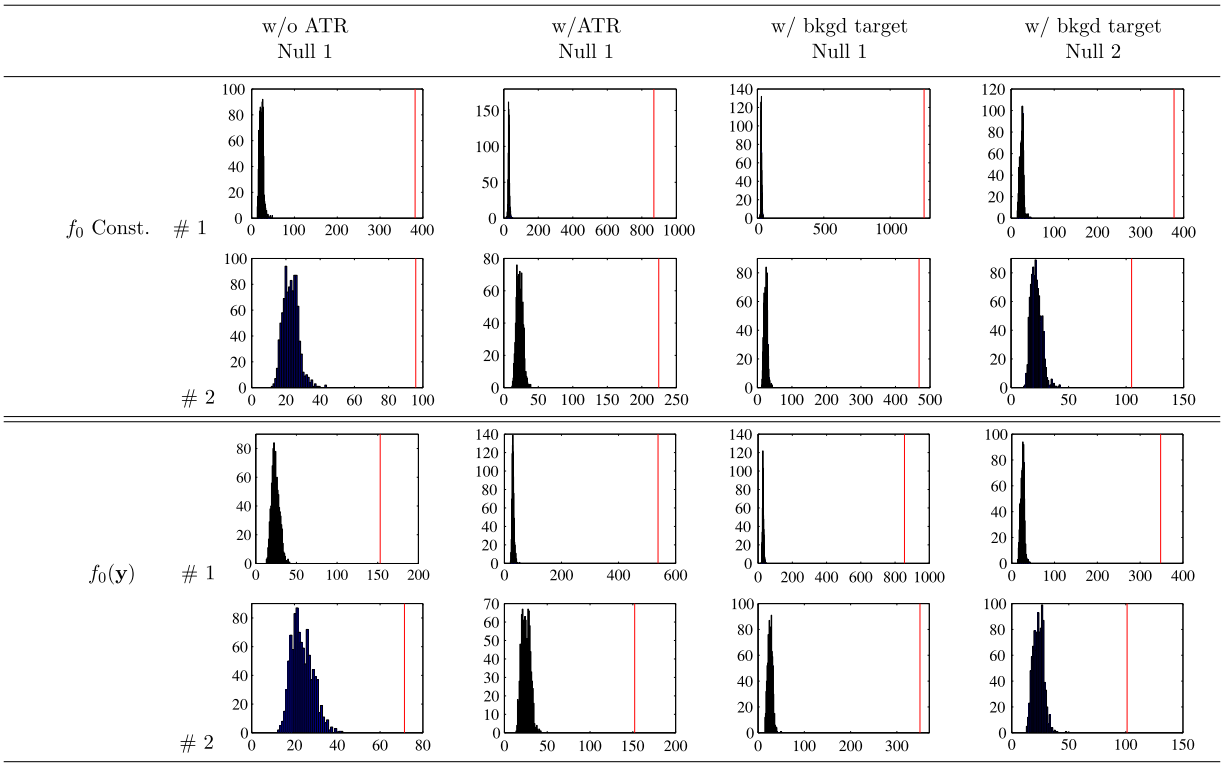

Fig. 18. Hypothesis testing on real datasets 1 ("# 1") and 2 ("# 2") under various model configurations. Each panel shows a histogram of samples under the null distribution of D along with the value of d^* (vertical red line) for each respective case. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

testing under all possible model configurations yields a rejection of the null hypothesis in both datasets, we accept the alternate hypothesis that there is a target line in both scenes.

7. Summary and discussion

To summarize our contribution, we have developed a statistical model for the problem of estimating a target-layer trajectory, a straight line, using spatial point clouds generated by ATR algorithms. These point clouds are characterized by excess clutter, making it challenging to detect target locations in them. We model target locations as realizations of a marked Poisson process, perturbed by a Gaussian observation noise, with marks provided by the ATR scores. Similarly, the clutter is modeled as a homogeneous, marked Poisson process on the observation domain. Then, we use optimization techniques to solve for MLE of model parameters and use them to test the presence of line in a given data. The results, demonstrated on both simulated and real target data, show a good performance in line estimation. We also extend the model to include situations where target points also come from a homogeneous Poisson process, in addition to the ones associated with the target-layer trajectory.

In terms of further generalizing this model, one can study trajectories that are not lines but have *known* geometry, for instance arcs, ellipses, polygons, etc. Since the current paper utilizes the polar representation of a straight line in defining the log-likelihood and its derivatives, one would need to express the quantities of interest in those relevant parameters, but the remaining optimization and inference framework should be similar. Another possibility is to allow *unknown* geometries for the trajectories. For example, one can assume that the trajectory comes from a fixed shape class but allow some variability within that class. The solution can come from active contour methods that initialize the trajectory arbitrarily and update coordinates of the contour iteratively within the desired shape class.

Appendix. Computing J and its partial derivatives

In this appendix we drop the use of boldface for vectors. The intensity of targets at the point $y \in \mathbb{R}^2$ is $\gamma\alpha_\sigma(y)$ with

$$\alpha_\sigma(y) = \int_L f_{\sigma^2}(y|s)\, ds,$$

where L is a line in $\mathbb{R}^2$, s denotes a point on this line, ds denotes Lebesgue measure on L (i.e., measure = length for intervals on L), and $f_t(y|\mu)$ is a bivariate normal density with mean μ and variance tI. If we rotate and translate the plane, we can

move the line L to the x-axis and the point y to a point $(0, b)$ on the y-axis. Here $b = d(y)$ is the minimum distance between y and L. Then the integral above becomes

$$\alpha_\sigma(y) = \frac{1}{2\pi\sigma^2}\int_{-\infty}^{\infty} \exp(-(x^2+b^2)/(2\sigma^2))\,dx = \frac{\exp(-b^2/(2\sigma^2))}{\sqrt{2\pi\sigma^2}},$$

so that

$$\alpha_\sigma(y) = \frac{\exp(-d(y)^2/(2\sigma^2))}{\sqrt{2\pi\sigma^2}}. \tag{A.1}$$

Let (r, ϕ) be the polar coordinates of the line L (i.e., the polar coordinates of that point on the line that is closest to the origin), and let $v = (\cos\phi, \sin\phi)$ be the unit vector perpendicular to the line L. Then $d(y) = |p(y)|$, where $p(y) = y \cdot v - r$, and we may re-write Eq. (A.1) as

$$\alpha_\sigma(y) = \varphi_{\sigma^2}(p(y)), \quad \text{where } \varphi_t(z) = \frac{e^{-z^2/(2t)}}{\sqrt{2\pi t}}. \tag{A.2}$$

(We shall use φ_t and Φ_t to denote the pdf and cdf of the normal distribution with variance t, and $\varphi \equiv \varphi_1$ and $\Phi \equiv \Phi_1$ for the pdf and cdf when the variance equals 1.) In what immediately follows, the variance σ^2 is more convenient than σ as an argument, and so we let $t = \sigma^2$ and define

$$A_t(y) = \alpha_{\sqrt{t}}(y) = \varphi_t(p(y)). \tag{A.3}$$

Let U be the fixed region in $\mathbb{R}^2$. We desire to compute

$$J \equiv J(r, \phi, t) = \int_U A_t(y)\,dy \tag{A.4}$$

and its gradient $(\frac{\partial J}{\partial r}, \frac{\partial J}{\partial \phi}, \frac{\partial J}{\partial t})$.

The bivariate normal density $f \equiv f_t(y|\mu)$ satisfies the heat equation

$$\frac{\partial f}{\partial t} = \frac{1}{2}\left(\frac{\partial^2 f}{\partial y_1^2} + \frac{\partial^2 f}{\partial y_2^2}\right) = \frac{1}{2}\nabla\cdot\nabla f$$

for all μ. Since $A_t(y)$ is a continuous superposition of such functions, it also satisfies the heat equation. Therefore, by the divergence theorem, we have

$$\frac{\partial}{\partial t}\int_U A_t(y)\,dy = \frac{1}{2}\int_U \nabla\cdot\nabla A_t(y)\,dy = \frac{1}{2}\oint (\nabla A_t(y))\cdot\hat{n}\,ds, \tag{A.5}$$

where $\hat{n}$ denotes the outward pointing unit normal on the contour surrounding U, and ds is an infinitesimal arc length. From (A.3) we obtain $\nabla A_t(y) = \varphi_t'(p(y))v$ so that

$$\frac{\partial J}{\partial t} = \frac{1}{2}\oint \varphi_t'(p(y))(v\cdot\hat{n})\,ds. \tag{A.6}$$

The desired function J is an integral (anti-derivative) of (A.6) with respect to t. It is easily verified that

$$\frac{\partial}{\partial t}\left(\Phi_t(z) - \frac{1}{2}\right) = \frac{1}{2}\varphi_t'(z) \quad \text{and} \quad \lim_{t\to\infty}\left(\Phi_t(z) - \frac{1}{2}\right) = 0,$$

so that, by differentiating inside the integral, we see that

$$J = \oint \left(\Phi_t(p(y)) - \frac{1}{2}\right)(v\cdot\hat{n})\,ds \tag{A.7}$$

satisfies (A.6). This definition also satisfies $J \to 0$ as $t \to \infty$ so that it is the correct choice of the anti-derivative. (Actually, one can replace $\frac{1}{2}$ in (A.7) by any constant without affecting the value of J.)

Up to this point, we have made no assumptions about the region U. Now we restrict ourselves to polygonal regions. Assume there are k vertices which (in counterclockwise order starting from an arbitrary vertex) are $y_1, y_2, \ldots, y_k$. Define $y_{k+1} = y_1$. Let $S_1, S_2, \ldots, S_k$ be the sides of the polygon, where S_i is the segment joining y_i and y_{i+1}. Let $\hat{n}_i$ be the outward pointing unit normal on side S_i, and $\widehat{ds}_i = (y_{i+1} - y_i)/|y_{i+1} - y_i|$ be the unit vector pointing along S_i. Let θ_i be the angle between $\widehat{ds}_i$ and the x-axis.

Then (A.7) can be written

$$J = \sum_{i=1}^{k}(v\cdot\hat{n}_i)\int_{S_i}(\Phi_t(p(y)) - 1/2)\;ds. \tag{A.8}$$

As one moves along segment S_i from y_i to y_{i+1}, the function $p(y)$ will increase as a linear function of the arc length with slope

$$\frac{\Delta d}{\Delta s} = \frac{p(y_{i+1}) - p(y_i)}{\text{length}(S_i)} = \cos(\phi - \theta_i) = v \cdot \widehat{ds}_i, \tag{A.9}$$

where $\phi - \theta_i$ is the angle between v and $\widehat{ds}_i$. Therefore,

$$\int_{S_i} (\Phi_t(p(y)) - 1/2)\ ds = \int_{p_i}^{p_{i+1}} (\Phi_t(p) - 1/2)\ dp \left(\frac{\Delta s}{\Delta d}\right), \tag{A.10}$$

where we define $p_i = p(y_i)$ for all i. This last integral has a closed form. Define

$$\begin{aligned}\psi_t(u) &= t\varphi_t(u) + u\left(\Phi_t(u) - \frac{1}{2}\right)\\ &= \sqrt{t}\ \varphi\left(\frac{u}{\sqrt{t}}\right) + u\left(\Phi\left(\frac{u}{\sqrt{t}}\right) - \frac{1}{2}\right).\end{aligned} \tag{A.11}$$

It is easy to verify that

$$\psi_t'(u) = \Phi_t(u) - \frac{1}{2},$$

so that

$$\int_{p_i}^{p_{i+1}} (\Phi_t(p) - 1/2)\ dp = \psi_t(p_{i+1}) - \psi_t(p_i),$$

and from (A.8), (A.9), and (A.10) we conclude

$$J = \sum_{i=1}^{k} \frac{(v \cdot \hat{n}_i)}{(v \cdot \widehat{ds}_i)} (\psi_t(p_{i+1}) - \psi_t(p_i)). \tag{A.12}$$

The i-th term in the summation (A.12) is undefined when the side S_i is parallel to the line L. In this case $v \cdot \widehat{ds}_i = 0$, and $p(y)$ is constant on S_i so that $p_{i+1} = p_i$. But then it is clear from (A.8) that the i-th term should be

$$\left(\Phi\left(\frac{p_i}{\sqrt{t}}\right) - \frac{1}{2}\right)(v \cdot \hat{n}_i) \times \text{length}(S_i). \tag{A.13}$$

For later reference we note that

$$\frac{(v \cdot \hat{n}_i)}{(v \cdot \widehat{ds}_i)} = \tan(\theta_i - \phi), \tag{A.14}$$

which follows from (A.9) and the fact that the angle between v and $\hat{n}_i$ is $\phi - (\theta_i - \pi/2)$ so that $v \cdot \hat{n}_i = \cos(\phi - \theta_i + \pi/2) = \sin(\theta_i - \phi)$.

The argument t has served its purpose, and we now re-define J and ψ as functions of $\sigma = \sqrt{t}$ by everywhere replacing $\sqrt{t}$ by σ. This leads to

$$J \equiv J(r, \phi, \sigma) = \sum_{i=1}^{k} \frac{(v \cdot \hat{n}_i)}{(v \cdot \widehat{ds}_i)} (\psi_\sigma(p_{i+1}) - \psi_\sigma(p_i)), \tag{A.15}$$

where

$$\psi_\sigma(u) \equiv \sigma\ \varphi\left(\frac{u}{\sigma}\right) + u\left(\Phi\left(\frac{u}{\sigma}\right) - \frac{1}{2}\right). \tag{A.16}$$

When computing (A.15), undefined terms in the summation are replaced by

$$\left(\Phi\left(\frac{p_i}{\sigma}\right) - \frac{1}{2}\right)(v \cdot \hat{n}_i) \times \text{length}(S_i). \tag{A.17}$$

Now we calculate the gradient of $J(r, \phi, \sigma)$. In calculating these partial derivatives, we use the facts

$$\frac{\partial \psi_\sigma(u)}{\partial u} = \Phi\left(\frac{u}{\sigma}\right) - \frac{1}{2} \quad \text{and} \quad \frac{\partial \psi_\sigma(u)}{\partial \sigma} = \varphi\left(\frac{u}{\sigma}\right), \tag{A.18}$$

which are easily verified from (A.16).

Differentiating (A.15) with respect to σ we easily obtain

$$\frac{\partial J}{\partial \sigma} = \sum_{i=1}^{k} \frac{(v \cdot \hat{n}_i)}{(v \cdot \widehat{ds}_i)} \left\{ \varphi\left(\frac{p_{i+1}}{\sigma}\right) - \varphi\left(\frac{p_i}{\sigma}\right) \right\}. \tag{A.19}$$

Undefined terms in the summation (A.19) are replaced by

$$\frac{-p_i}{\sigma^2} \varphi\left(\frac{p_i}{\sigma}\right) (v \cdot \hat{n}_i) \times \text{length}(S_i), \tag{A.20}$$

which is the derivative of (A.17) with respect to σ.

Recalling that $p_i = y_i \cdot v - r$ and differentiating (A.15) with respect to r, we similarly obtain

$$\frac{\partial J}{\partial r} = -\sum_{i=1}^{k} \frac{(v \cdot \hat{n}_i)}{(v \cdot \widehat{ds}_i)} \left\{ \Phi\left(\frac{p_{i+1}}{\sigma}\right) - \Phi\left(\frac{p_i}{\sigma}\right) \right\}. \tag{A.21}$$

Undefined terms in the summation (A.21) are replaced by

$$\frac{-1}{\sigma} \varphi\left(\frac{p_i}{\sigma}\right) (v \cdot \hat{n}_i) \times \text{length}(S_i), \tag{A.22}$$

which is the derivative of (A.17) with respect to r.

Computing the partial of J with respect to ϕ is more difficult since $v = (\cos(\phi), \sin(\phi))$ is involved in all parts of (A.15). Using (A.14) to re-write (A.15) as

$$J = \sum_{i=1}^{k} \tan(\theta_i - \phi)(\psi_\sigma(p_{i+1}) - \psi_\sigma(p_i))$$

and then differentiating with respect to ϕ using the product rule leads to

$$\begin{aligned}
\frac{\partial J}{\partial \phi} &= \sum_{i=1}^{k} \tan(\theta_i - \phi) \left[\left(\Phi\left(\frac{p_{i+1}}{\sigma}\right) - \frac{1}{2} \right) \left(y_{i+1} \cdot \frac{\partial v}{\partial \phi} \right) - \left(\Phi\left(\frac{p_i}{\sigma}\right) - \frac{1}{2} \right) \left(y_i \cdot \frac{\partial v}{\partial \phi} \right) \right] \\
&\quad - \sum_{i=1}^{k} \sec^2(\theta_i - \phi) \left[\psi_\sigma(p_{i+1}) - \psi_\sigma(p_i) \right] \\
&= \sum_{i=1}^{k} \frac{(v \cdot \hat{n}_i)}{(v \cdot \widehat{ds}_i)} \left[\left(\Phi\left(\frac{p_{i+1}}{\sigma}\right) - \frac{1}{2} \right) (y_{i+1} \cdot w) - \left(\Phi\left(\frac{p_i}{\sigma}\right) - \frac{1}{2} \right) (y_i \cdot w) \right] \\
&\quad - \sum_{i=1}^{k} \frac{1}{(v \cdot \widehat{ds}_i)^2} \left[\psi_\sigma(p_{i+1}) - \psi_\sigma(p_i) \right],
\end{aligned} \tag{A.23}$$

where

$$\frac{\partial v}{\partial \phi} = (-\sin(\phi), \cos(\phi)) \equiv w$$

is a unit vector perpendicular to v.

As usual, when S_i and L are parallel, the i-th term in the above formula is undefined. Unfortunately, in this case the correct answer is not (quite) obtained by simply differentiating (A.17) with respect to ϕ. To get a correct answer, we differentiate the i-th term in the general expression (A.8) with respect to ϕ, and then evaluate the result when S_i and L are parallel, in which case $v = \pm\hat{n}_i$, $w = \pm\widehat{ds}_i$ (with the same choice of sign), and $w \cdot \hat{n}_i = 0$. For convenience, we assume $v = \hat{n}_i$ and $w = \widehat{ds}_i$; the choice does not affect the answer. The calculation is as follows:

$$\begin{aligned}
&\frac{\partial}{\partial \phi} \left[(v \cdot \hat{n}_i) \int_{S_i} (\Phi_t(p(y)) - 1/2)\ ds \right] \\
&= \left(\frac{\partial v}{\partial \phi} \cdot \hat{n}_i \right) \int_{S_i} (\Phi_t(p(y)) - 1/2)\ ds + (v \cdot \hat{n}_i) \int_{S_i} \frac{\partial}{\partial \phi} (\Phi_t(p(y)) - 1/2)\ ds \\
&= \left(\frac{\partial v}{\partial \phi} \cdot \hat{n}_i \right) \int_{S_i} (\Phi_t(p(y)) - 1/2)\ ds + (v \cdot \hat{n}_i) \int_{S_i} \varphi_t(p(y)) \left(y \cdot \frac{\partial v}{\partial \phi} \right) ds \\
&\quad \text{(Now evaluate when } S_i \text{ and } L \text{ are parallel.)} \\
&= 0 + (\hat{n}_i \cdot \hat{n}_i) \varphi_t(p_i) \int_{S_i} \left(y \cdot \widehat{ds}_i \right) ds
\end{aligned}$$

$$
\begin{aligned}
&= \varphi_t(p_i)\,\frac{1}{2}\left[\left(y_{i+1}\cdot\widehat{ds}_i\right)^2-\left(y_i\cdot\widehat{ds}_i\right)^2\right]\\
&= \varphi_t(p_i)\,\frac{1}{2}\left[\left(y_{i+1}\cdot\widehat{ds}_i\right)+\left(y_i\cdot\widehat{ds}_i\right)\right]\left[\left(y_{i+1}\cdot\widehat{ds}_i\right)-\left(y_i\cdot\widehat{ds}_i\right)\right]\\
&= \varphi_t(p_i)\left[\frac{1}{2}\left(y_i+y_{i+1}\right)\cdot\widehat{ds}_i\right]\times\text{length}(S_i)\\
&= \frac{1}{\sigma}\varphi\left(\frac{p_i}{\sigma}\right)\left[\frac{1}{2}\left(y_i+y_{i+1}\right)\cdot w\right]\left(v\cdot\hat{n}_i\right)\times\text{length}(S_i).
\end{aligned}
$$

The changes introduced in the last line are to make it more closely parallel earlier expressions. This last line gives the replacement for the i-th term in (A.23) when it is undefined.

References

Agarwal, S., Ramachandran, H., Kummamuru, S., Mitchell, O.R., 2002. Algorithms and architecture for airborne minefield detection. Proc. SPIE 4742, 96–107.

Bryner, D., Huffer, F., Srivastava, A., Tucker, J.D., 2014. Underwater mine detection in clutter data using spatial point process models. IEEE J. Ocean. Eng. Rev..

Byers, S., Raftery, A.E., 1998. Nearest-neighbor clutter removal for estimating features in spatial point processes. J. Amer. Statist. Assoc. 93 (442), 577–584.

Chandra, V., Elgar, S., Nguyen, A., 2002. Detection of mines in acoustic images using higher order spectral features. IEEE J. Ocean. Eng. 27 (3), 610–618.

Cressie, N., Lawson, A.B., 1998. Bayesian hierarchical analysis of minefield data. Proc. SPIE 3392, 930–940.

Fischler, M.A., Bolles, R., 1981. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM 24 (6), 381–395.

Isaacs, J., Tucker, J., 2011. Diffusion features for target specific recognition with synthetic aperture sonar raw signals and acoustic color. In: IEEE International Conference on Pattern Recognition and Computer Vision Workshops, pp. 27–32.

Lake, D.E., 1998. Families of statistics for detecting minefields. Proc. SPIE 3392, 963–974.

Lake, D.E., Sadler, B., Casey, S., 1997. Detecting regularity in minefields using collinearity and a modified euclidean algorithm. Proc. SPIE 3079, 500–507.

Ratches, J.A., 2011. Review of current aided/automatic target acquisition technology for military target acquisition tasks. SPIE J. Opt. Eng. 50 (7), online.

Soumekh, M., 1999. Synthetic Aperture Radar Signal Processing. Wiley, New York.

Stack, J., 2011. Automation for underwater mine recognition: current trends and future strategy. Proc. SPIE 8017, 1–21.

Su, J., Huffer, F., Srivastava, A., 2013. Detection, classification and estimation of shapes in 2D and 3D point clouds. Comput. Statist. Data Anal. 58, 227–241.

Trang, A., Agarwal, S., Broach, J.T., Smith, T., 2008. Exploiting "mineness" for scatterable minefield detection. Proc. SPIE 6953, 1–12.

Trang, A., Agarwal, S., Broach, J.T., Smith, T., 2011. Validating spectral spatial detection based on mmpp formulation. Proc. SPIE 8017, 1–12.

Tucker, J., Azimi-Sadjadi, M., 2011. Coherence-based underwater target detection from multiple sonar platforms. IEEE J. Ocean. Eng. 36 (1), 37–51.

Walsh, D.C., Raftery, A.E., 2002. Detecting mines in minefields with linear characteristics. Technometrics 44 (1), 34–44.

Walsh, D.C., Raftery, A.E., 2005. Classification of mixtures of spatial point processes via partial Bayes factors. J. Comput. Graph. Statist. 14 (1), 139–154.