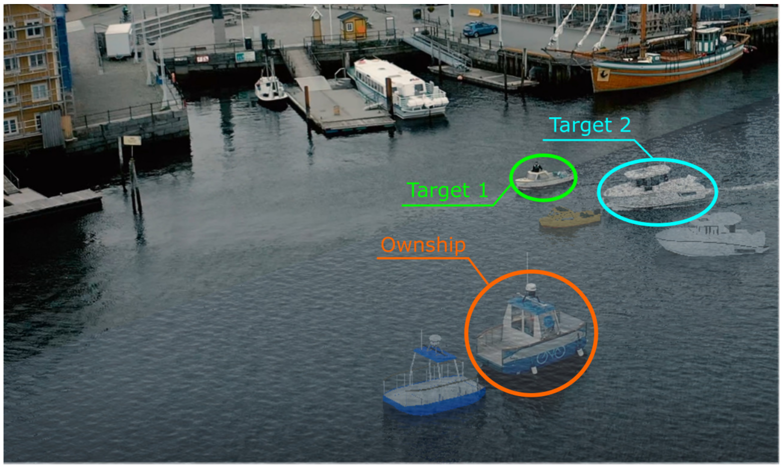
Hellinger Metrics for Validating High Fidelity Simulators using Target Tracking

Kjetil Vasstein[1], Øystein Kaarstad Helgesen[2], and Edmund Førland Brekke[3],

Norwegian University of Science and Technology (NTNU),
Department of Engineering Cybernetics,
O. S. Bragstads plass 2D, 7032 Trondheim, Norway

[1] kjetil.vasstein@ntnu.no,
[2] oystein.k.helgesen@ntnu.no,
[3] edmund.brekke@ntnu.no



Abstract. To achieve autonomy at sea, it is believed simulations will be essential in testing, analysing and verifying autonomous systems due to the scarcity and high cost of obtaining real data for all relevant scenarios. This reliance on simulation raises the question on how much synthetic data can be trusted, especially from sensor data such as lidars, radars and cameras. Methods for validating specific sensor models exists, however these are often focusing on perceptional differences without considering the sensors impact on the autonomy's situational awareness. In this paper we make an attempt to analyse this using a JIPDA target tracker, by comparing its performance on real and synthetic lidar data with various Hellinger metrics for Bernoulli and multi-Bernoulli multi-target densities. Our work showcases a method that quantifies sensor fidelity of synthetic data based on tracker performance, a step towards building trust to simulators targeted at validating autonomy systems.

Keywords: Simulation validation, Multi-target tracking, Multi-Bernoulli, Hellinger distance, Finite set statistics, Csiszár information functional, Maritime autonomy, Situational awareness

List of Symbols

$\mathcal{I}$	Csiszár information functional
$\mathcal{N}(x;\, \boldsymbol{\mu},\, \mathbf{P})$	Multivariate Gaussian distribution
w_k	Multivariate Gaussian measurement noise
v_k	Multivariate Gaussian process noise
$\mathcal{C}$	Kernel to Csiszár information functional
p	Kinematic probability density function
$NEES$	Normal estimation error squared
f_z	Sensor model for tracker
$\mathbf{D}$	Covariance matrix for Gaussian Hellinger metric
$\mathbf{C}$	Covariance matrix for Gaussian product
$\mathbf{P}$	Covariance matrix for multivariate Gaussian
$\mathbf{J}$	The Jacobian matrix of the polar to Cartesian conversion of the measurement
$\mathbf{R}$	Covariance for measurement noise in JIPDA
$\mathbf{G}$	Process noise input matrix
$\mathbf{Q}_k$	Covariance of process noise
$\mathbf{O}$	Concatenated covariance matrix of Gaussian products
$\mathbf{A}$	Transition matrix for continuous time
$\mathbf{F}$	Transition matrix for discrete time
ξ	Cardinality of X
d	Covariance difference in Gaussian Hellinger metric
n	Dimentionality of X, i.e. length of $\boldsymbol{x}$
x	Realised vector element of $\boldsymbol{x}$
q	Existence probability
i	Index number
k_ξ	Normalisation constant for concatenated Gaussian densities
s	Number of Bernoulli components
k	Time increment
$g_{A,B}$	Existence weight for active target occurrences in dataset A and B
X	The random finite set of potential objects at each time instance k
σ	Bernoulli components of active targets
$A,\ B$	Subscript notation for dataset A and B
$\mathit{\Xi}$	The space of the random finite set X
$\boldsymbol{u}$	Vector estimate of concatenated Gaussian densities
$\boldsymbol{x}$	Vector realisation of X
$\boldsymbol{\mu}$	Estimate of $\boldsymbol{x}$
$\boldsymbol{n}$	Random Gaussian white noise vector
$\boldsymbol{v}$	Estimate vector for product of Gaussian densities
$f_{\mathit{\Xi}}$	Multi-density function for an arbitrary space $\mathit{\Xi}$
$a,\ b$	Simplified terms in the Multi-Bernoulli Multi Denisty Function

1 Introduction

High fidelity sensor simulations is playing an ever increasing role in the development of autonomous vehicles. Especially in the machine learning community the use of simulation as a substitution for real world testing and training gives the developers a source for variable, accurate and unlimited data. This has the potential to give better generalisations and bigger test scopes which is required to guarantee safe and reliable autonomous operations. Having high fidelity is of benefit here as it is thought to help transition the artificial intelligence (AI) algorithms from simulations to real world applications.

However, the simulations' execution time is dependent on the level of fidelity, meaning there must be a balance between data quantity and quality. Unfortunately, judging fidelity of simulators targeting autonomy such as Carla [7], Gemini [14] and AirSim [12] is often done by intuition on what "looks" more real. This is despite the driver, captain or pilot in autonomy cases being a machine. This can lead us astray when improving simulation models and give us false hope for the final deployment as what looks more real for us humans does not necessarily imply the same for the AI [6, p. 3171]. To get an optimal relation between simulation and autonomy, we need to answer how fidelity affects the AI's performance. This will be the goal of any simulation framework that promises to deliver on high data quantity at the right quality. Having adequate metrics that measures fidelity relative to its impact on autonomy systems will help to establish validation techniques that can benchmark the simulation performance, moreover help ensuring the simulation development goes in the right direction.

Transferring autonomy systems developed in simulation to reality is a particular case of domain adaptation [6, p. 3]. Here the simulation the autonomous agent is trained and tested in is defined as the source domain, and its real world deployment defined as the target domain. There are several known methods that helps to improve this transition. Domain randomisation is a technique where the simulator uses procedural generation to vary textures, content and situations to increase the chance for the AI to perceive the real world as yet another variation [13]. Augmenting the source domain to better reassemble the target domain from generative adversarial networks (GAN) [11] is yet another technique. However most, of these methods rely on machine learning concepts to make autonomy deployable in target domains. Since machine learning uses a black box modelling concept that is unexplainable and unpredictable, it is questionable if these techniques can be considered safe and viable for validation. As an example, the use of CycleGAN [19] to improve the quality of synthetic images showed no performance increase for the autonomy despite the images looking more realistic [6]. If this problem applies in general to machine learning is speculative, as reasoning about black box systems is far from trivial. This gives motivation for a validation approach that relies less on black box systems to be more explainable.

Instead of focusing on what the AI perceives, one could instead focus on what the AI understands of the situation, i.e., to study the Situational Awareness (SITAW) of the AI. The approaches we have discussed so far and which is fairly popular is end-to-end learning. Here SITAW is incorporated into a black box system often created through machine learning using artificial neural networks. The input to these systems are raw sensor data while the output may be signals to actuators, meaning that SITAW may actually not exist in any meaningful sense. Without any internal insight of the black box system, validation of these systems are hard to do without testing the whole system. A more validation friendly approach is to modularize the system as much as possible, so that the pipeline can be divided into components using machine learning and explainable model-based techniques. Here SITAW plays a role in the higher modularisation scheme in addition to containing modules of its own.

Core tasks in a SITAW system are detection and tracking, which can be solved by means of Bayesian filters. The role of detectors is to give information about potential surrounding entities, i.e., detections. The tracker on the other hand, consists (among others) of filters with the purpose of ensembling the present detections with previous beliefs about the targets to establish tracks on them. When assuming that the targets' measurements, processes and initialisations follow Gaussian models, the Kalman filter or its extended version can be used as a closed form solution of the general Bayesian filter. Here target states are estimated based on noisy measurements originating from either targets or from false alarms. This requires the tracker to also associate measurements to targets which can be done in several ways. In the Probabilistic Data Association (PDA) [3] family of tracking methods, individual measurements are used to update target states based on the likelihood of it originating from the target or from clutter. The Joint Integrated PDA (JIPDA) [10] is a multi-target extension of the Integrated PDA [9] which extends PDA with estimates of target existence probability.

A concept similar to domain adaptation for trackers is filter tuning. Here various metrics help guide the developer to tune the tracker towards the final deployment. For single-target / single-sensor analysis, metrics could either be measuring a point-point distance, or probability distribution distance between what is estimated by a tracker and what is considered to be the *ground truth*. The current state of the art in the tracking field is however in multi-target tracking, where *Finite-Set Statistics* (FISST) have been responsible for several innovations. Among these are the creation of metrics better suited for evaluating multi-target trackers, where the *Csiszár's Information Functionals* is an example of a mathematical framework responsible for several of them [18]. Albeit this was originally intended for developing performance, efficiency and robustness metrics to evaluate tracker to tracker, the Csiszár information functional more generally compares probability distributions. This have found use cases in fields outside the tracking community, where the functional have among others been used in domain adaptations for GAN [2].

Other attempts of measuring autonomy performance have also been conducted in recent years, among which the robotic platform RoboThor have targeted how well robots can adapt to real world situations when being trained in synthetically recreated environments [6]. Here the agents performance is judged by its navigational performance using *Success Rate* and *Success weighted by Path Length* for each completed task. Results shows that even with almost pixel perfect reconstructed camera data used for navigation, there can be a significant difference in the agent's performance between simulation and reality. However, since the metrics mainly measures the navigational performance, moreover relies on end-to-end testings for obtaining results, this makes it hard to tell where in the autonomy pipeline the simulation and real world performance diverges.

A similar attempt of comparing simulation to reality with the use of synthetic reproduction have also been done for SITAW [15]. Here an autonomous ferry [4] including a digital twin representation [14] was used to gather datasets running through a detection and JIPDA tracking pipeline on both synthetic and real sensor data, comparing the trackers Gaussian posteriors using a Hellinger distance as a performance metric (Figure 1). One of the benefits of this was the ability to analyse arbitrary sensor data that could individually and collectively be studied for its impact on the tracker. In addition, the metric measures a particular portion of the autonomy pipeline rather the full end-to-end performance, making it both fast and specific of what it is measuring in the autonomy. However, the proposed Hellinger metric only considered single target kinematics with no attention to the trackers existence probability moreover track associations between synthetic and real data. The full output of a JIPDA have in contrary been shown to be a *multi-Bernoulli multi-object density function* (MBMDF) [16], which have also been confirmed by recent studies of the *Visual Interacting Multiple Model* JIPDA (VIMMJIPDA) extension [5].

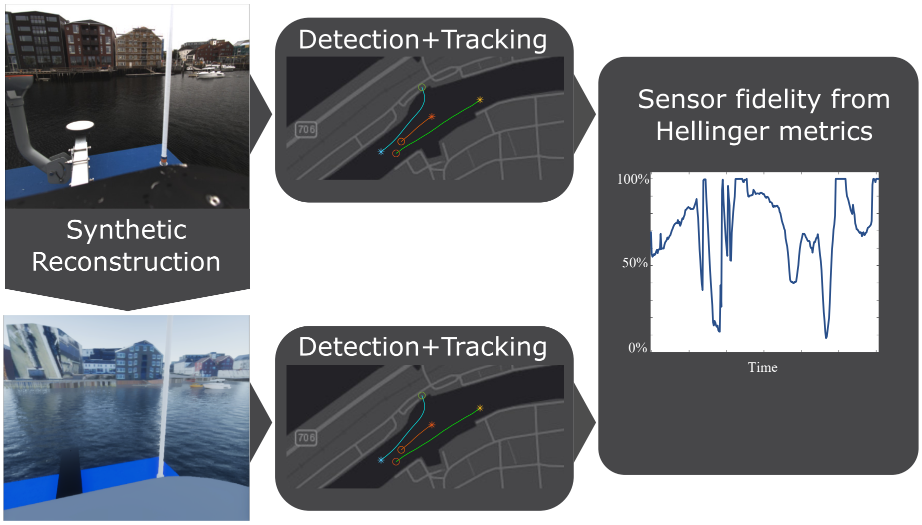


Fig. 1: Pipeline description

In this paper we will extend the attempt of comparing tracker performance on synthetic and real data using the same dataset as [15] focusing on its lidar data. We will include Bernoulli and multi-Bernoulli existence models by using Csiszár information functionals to facilitate for both single and multi-target cases. The new metrics will generalise and be compared to the previous Hellinger attempts in [15] by evaluating simulation performance with respect to reality, the effects of existence probabilities, and the metrics ability to measure tracker performance. This analysis can be used to obtain perspectives on 1) the simulations validity such as sensor fidelity, 2) scenario reproduction, and 3) the tracker's indifference or sensitiveness to simulated data with respect to reality.

We begin with outlining core concepts and definitions from FISST in Section 2 before detailing each step in the pipeline description (Figure 1) in the subsequent sections. Here we begin defining and deriving Hellinger metrics, before venturing into the JIPDA tracker in Section 4 where data is being processed to estimates, co-variances and existence probabilities of target entities in the datasets. This is followed by describing the synthetic and real dataset in Section 5. In Section 6 we go through some of the findings when comparing the datasets with the various performance metrics before we have a discussion in Section 7. Finally, we do a summary and conclude the paper with suggestions of future work in Section 8.

2 Finite-Set Statistics (FISST) for Metric Constructions

In order for target trackers to be considered viable, filter tuning is a necessary step in any research and design process. This relies on performance metrics, often based on statistical properties and assumptions of the filter. For the JIPDA tracker, targets are assumed to be represented as multivariate Gaussian distributions:

$$\mathcal{N}(\boldsymbol{x};\, \boldsymbol{\mu},\, \mathbf{P}) := \frac{1}{(2\pi)^{\frac{n}{2}}|\mathbf{P}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\mathbf{P}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right) \tag{1}$$

for a vector $\boldsymbol{x}$ with n-dimensions subjected to the expectation value $\boldsymbol{\mu}$ and covariance matrix $\mathbf{P}$. Relying on the Gaussian distribution allows us to check statistical metrics that must be in place for the filter to be considered viable. Among these are the *Normalised Estimation Error Squared* (NEES):

$$\mathit{NEES}(\boldsymbol{x}, \boldsymbol{\mu}, \mathbf{P}) := (\boldsymbol{x}-\boldsymbol{\mu})^T\mathbf{P}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}). \tag{2}$$

This measures the distance between a point $\boldsymbol{x}$ with a distribution consisting of an estimate $\boldsymbol{\mu}$ and covariance $\mathbf{P}$. In filter tuning for single target tracking, $\boldsymbol{x}$ is thought to be the ground truth giving us NEES values we interpret as *filter confidence*. The name *ground truth* refers to high accuracy measurements of target states that will be compared to the less perfect filter estimates. This becomes a comparison between two datasets we will note as A and B.

However, when the datasets are both distributions as is the case when comparing tracker outputs, a different approach is needed to handle additional information such as having two sets of covariance matrices instead of just one as in (2). In addition, distributions coming from the tracker is often accompanied by existence probabilities, which requires special treatment for single and multi-target cases.

In this section we will introduce mathematics from FISST that can be used to handle these concerns. We begin with defining random finite sets before we continue with *multi-object density functions* (MDF) for cases of Bernoulli and *multi-Bernoulli* (MB) distributions for existence probabilities. This is followed by defining Csiszár's information functionals from which special cases of Hellinger metrics is further derived and analysed in Section 3.

2.1 Random finite sets

We define X to be the set of potential objects at each time instance k: $X = \{\boldsymbol{x}_1 , ..., \boldsymbol{x}_\xi\}$, with each vector containing states of the object $\boldsymbol{x} = [x_1, ..., x_n]^T$. ξ is the cardinality of the set which represents the number of object realizations: $|X| = \xi$.

2.2 Multi-object Density Functions (MDF)

A MDF is written as $f_\Xi(X)$ where the subscript Ξ notes the space containing the RFS X.

Set integral. For an MDF the set integral is defined to be [8, p. 62]:

$$\int f_\Xi(X)\delta X := \sum_{\xi\geq 0} \frac{1}{\xi!} \int_{-\infty}^{\infty} f_\Xi(\{(\boldsymbol{x}_1) , ..., (\boldsymbol{x}_\xi)\}) d\boldsymbol{x}_1 ... d\boldsymbol{x}_\xi \tag{3}$$

Bernoulli. The simplest MDF to choose for a tracker is the Bernoulli distribution [8, p. 100]:

$$f_\Xi(X) := \begin{cases} 1 - q_\Xi \qquad \text{when } \ X = \{\emptyset\} \\ q_\Xi p_\Xi(\boldsymbol{x}) \qquad \text{when } \ X = \{\boldsymbol{x}\} \\ 0 \qquad \text{when } \ |X| \geq 2 \end{cases} , \tag{4}$$

where q denotes the existence probability of a target and $p_\Xi(\boldsymbol{x})$ the kinematic probability density function.

Multi-Bernoulli (MB). For handling multiple targets, the Bernoulli distribution is generalised to the MB distribution. Where Bernoulli can handle a maximum of one target, MB can handle multiple targets in a total of s Bernoulli components each with a unique existence probability. The MBMDF can be written as follows [8, p. 101]:

$$
\begin{aligned}
f_{\Xi}(X) &:= \sum_{\sigma \subseteq 1:s} \prod_{i=1}^{s} a_{\Xi}^{i} \prod_{i=1}^{\xi} b_{\Xi}^{\sigma(i)}, \\
a_{\Xi}^{i} &:= \left(1 - q_{\Xi}^{i}\right), \\
b_{\Xi}^{\sigma(i)} &:= \frac{q_{\Xi}^{\sigma(i)} p_{\Xi}^{\sigma(i)}(\boldsymbol{x}_i)}{1 - q_{\Xi}^{\sigma(i)}},
\end{aligned}
\tag{5}
$$

where a_{Ξ}^{i} and $b_{\Xi}^{\sigma(i)}$ are used for simplifying notations later in Section 3.3. The Bernoulli components of active targets are selected by a mapping σ from $\{1, \dots \xi\}$ to $\{1, \dots s\}$ where the total quantity of active targets becomes $|\sigma| = \xi$.

Special case of Bernoulli and MB. If there is only a single Bernoulli component, then the Bernoulli distribution $f_{\mathcal{B}}(X)$ is a special case of the MB distribution $f_{\mathcal{MB}}(X)$:

$$
f_{\mathcal{B}}(X) = f_{\mathcal{MB}}(X) \qquad \text{when} \;\; s = 1. \tag{6}
$$

2.3 Csiszár Information Functionals

Csiszár information functionals are defined as [8, p. 154]:

$$
\mathcal{I}_{\mathcal{C}}(f_A\,;\,f_B) := \int \mathcal{C}\left(\frac{f_A(X)}{f_B(X))}\right) f_B(X)\delta X, \tag{7}
$$

where f_A and f_B are MDF posteriors for multi-object trackers. If $\mathcal{C}$ is a *convex kernel* then $\mathcal{I}_{\mathcal{C}}(f_A\,;\,f_B) \geq 0$ where equality occurs only if $f_A = f_B$ almost everywhere.

3 Hellinger Performance Metrics

A special case of Csiszár's information functionals can be derived by choosing the kernel $\mathcal{C}(x) = \frac{1}{2}(\sqrt{x} - 1)^2$, giving us a normalised Hellinger information functional that can be used to derive various Hellinger metrics [8, p. 155]:

$$
\mathcal{I}_{\mathcal{H}}(f_A\,;\,f_B) = 1 - \int \sqrt{f_A(X) f_B(X)}\delta X. \tag{8}
$$

We will begin by using this to derive the conventional Hellinger metric. This will then be generalised to consider the case of single-target existence, before we end the section with our most generic metric that handles the case for multi-target tracking.

3.1 Hellinger distance for Gaussian distributions

For Hellinger distances of single valued functions, the Hellinger functional becomes a normal Hellinger distance:

$$\mathcal{I}_{\mathcal{H}}(f_A\,;\,f_B) = \mathcal{I}_{\mathcal{H}}\left(p_A\,;\,p_B\right) = 1 - \int \sqrt{p_A(\boldsymbol{x})p_B(\boldsymbol{x})}dy. \tag{9}$$

Where p_A and p_B are kinematic probability density functions for each respective dataset. If the distributions are in additional Gaussian, the term takes the special form [1, p. 6]:

$$\begin{aligned}\mathcal{I}_{\mathcal{H}}\left(\mathcal{N}_A\,;\,\mathcal{N}_B\right) &= \sqrt{1 - d \times \exp\left\{-\frac{1}{8}\mathit{NEES}(\boldsymbol{x}_A, \boldsymbol{x}_B, \mathbf{D})\right\}},\\ d &:= \sqrt{\frac{\sqrt{|\mathbf{P}_A||\mathbf{P}_B|}}{|\mathbf{D}|}},\\ \mathbf{D} &:= \frac{\mathbf{P}_A + \mathbf{P}_B}{2},\end{aligned} \tag{10}$$

where d gives information about the co-variance difference and the exponential term of the bias, both between the value 0 and 1. The united covariance between the distributions is noted as $\mathbf{D}$. Note we use *NEES* here to reuse notation, not to draw parallels to its other known properties in the tracking community.

3.2 Bernoulli Hellinger distance for Gaussian distributions

It can be shown that the Bernoulli case of the Hellinger functional can be expressed as:

$$\begin{aligned}\mathcal{I}_{\mathcal{B},\mathcal{H}}(f_A\,;\,f_B) &= 1 - \int \sqrt{f_A(X)f_B(X)}dx\\ &= 1 - \sqrt{q_A q_B} - \sqrt{(1-q_A)(1-q_B)} + \sqrt{q_A q_B}\mathcal{I}_{\mathcal{H}}.\end{aligned} \tag{11}$$

Where q_A and q_B are existence probabilities for the target existing in each respective datasets. If we set the existence probabilities to 1, most terms cancels out and we are left with the conventional Hellinger distance, i.e., Bernoulli Hellinger is a generalisation of Hellinger.

If the distributions are Gaussian, (10) can be used for an explicit solution for $\mathcal{I}_{\mathcal{H}}$. Otherwise, (9) must be handled either analytically or numerically for $\mathcal{I}_{\mathcal{H}}$

3.3 MB-Hellinger distance for Gaussian distributions

$$\mathcal{I}_{\mathcal{MB},\mathcal{H}}(f_A\,;\,f_B) = 1 - \int \sqrt{f_A(X)f_B(X)}\delta X.$$

In the MB case the MDF's are MB (5), where (3) expresses the integral form. The product of the two MDF's can be written as:

$$
\begin{aligned}
f_A(X)f_B(X) &= \sum_{\sigma_A \subseteq 1:s_A} \prod_{i=1}^{s_A} a_A^i \prod_{i=1}^{\xi} b_A^{\sigma_A(i)} \sum_{\sigma_B \subseteq 1:s_B} \prod_{i=1}^{s_B} a_B^i \prod_{i=1}^{\xi} b_B^{\sigma_B(i)} \\
&= \sum_{\sigma_A \subseteq 1:s_A} \sum_{\sigma_B \subseteq 1:s_B} \prod_{i=1}^{s_A} a_A^i \prod_{i=1}^{s_B} a_B^i \prod_{i=1}^{\xi} b_A^{\sigma_A(i)} b_B^{\sigma_B(i)},
\end{aligned}
$$

Since a single MB is non zero only if the cardinality is less than or equal the amount of track instances, when combining two MB we get the following criteria: $|\sigma_A| = |\sigma_B| = \xi$, i.e we only need to sum the minimum number of track instances found in one of the datasets since rest will be zero. Given that the probability density functions are Gaussian, the last product can be written as follows:

$$
\begin{aligned}
b_A^{\sigma_A(i)} b_B^{\sigma_B(i)} &= g_{A,B} \mathcal{N}(\boldsymbol{x};\, \boldsymbol{\mu}_A^{\sigma_A(i)},\, \mathbf{P}_A^{\sigma_A(i)}) \mathcal{N}(\boldsymbol{x};\, \boldsymbol{\mu}_B^{\sigma_B(i)},\, \mathbf{P}_B^{\sigma_B(i)}) \\
&= g_{A,B} \mathcal{N}(\boldsymbol{\mu}_A^{\sigma_A(i)};\, \boldsymbol{\mu}_B^{\sigma_B(i)},\, \mathbf{P}_A^{\sigma_A(i)} + \mathbf{P}_B^{\sigma_B(i)}) \mathcal{N}(\boldsymbol{x};\, \boldsymbol{v}^i,\, \mathbf{C}^i),
\end{aligned}
$$

where the terms are defined as:

$$
\begin{aligned}
g_{A,B} &:= \frac{q_A^{\sigma_A(i)} q_B^{\sigma_B(i)}}{\left(1 - q_A^{\sigma_A(i)}\right)\left(1 - q_B^{\sigma_B(i)}\right)}, \\
\boldsymbol{v}^i &:= \mathbf{C}^i \left((\mathbf{P}_A^{\sigma_A(i)})^{-1} \boldsymbol{\mu}_A^{\sigma_A(i)} + (\mathbf{P}_B^{\sigma_B(i)})^{-1} \boldsymbol{\mu}_B^{\sigma_B(i)} \right)^{-1}, \\
\mathbf{C}^i &:= \left((\mathbf{P}_A^{\sigma_A(i)})^{-1} + (\mathbf{P}_B^{\sigma_B(i)})^{-1} \right)^{-1}.
\end{aligned}
$$

Finally, the products of the Gaussian densities can be concatenated as follows:

$$
\prod_{i=1}^{\xi} \mathcal{N}(\boldsymbol{x};\, \boldsymbol{v}^i,\, \mathbf{C}^i) = k_\xi \mathcal{N}(\boldsymbol{x};\, \boldsymbol{u},\, \mathbf{O}),
$$

$$
\boldsymbol{u} := \begin{bmatrix} \boldsymbol{v}^1 \\ \vdots \\ \boldsymbol{v}^\xi \end{bmatrix}, \mathbf{O} := \begin{bmatrix} \mathbf{C}^1 & & \\ & \ddots & \\ & & \mathbf{C}^\xi \end{bmatrix}, k_\xi := \frac{|\mathbf{O}|^{\frac{1}{2}}}{\prod_{i=1}^{\xi} |\mathbf{C}^i|^{\frac{1}{2}}}.
$$

Importance sampling. In comparison to Hellinger and the Bernoulli Hellinger metrics, the integral term in MB-Hellinger is not trivial to solve explicitly due to the square root of sums. Because of this we use importance sampling to approximate the integral. We choose our importance density to be the normalised version of the Gaussian mixture $f_A(X)f_B(X)$, with its co-variances inflated by more than 2 to compensate for the square root (we have used a value of 3.6). This gives us enough coverage of the sampling area to estimate the integral.

4 JIPDA Tracker

In this section we give a brief introduction to the JIPDA tracker used in this work as well as the sensor pipeline used to process lidar data.

4.1 Lidar detection pipeline

Sensor data from the lidar is natively supplied as a point cloud and requires multiple processing steps before it can be utilized by the tracker.

Land filtering. Active sensors such as lidar will, if in range, yield positive returns from non-target entities such as land and buildings. If these detections are used directly in the tracker without processing they might induce a large amount of false tracks. To combat this, map based filtering is employed to remove unwanted lidar returns. Based on data from the Norwegian Mapping Authority, a binary occupancy grid is generated for the operating area. The lidar point cloud is then projected onto this map and any point falling within a land cell is removed.

Clustering. Another issue with lidar sensor data is that the sensor resolution is high enough to yield multiple returns from a single target. This violates the assumption that only one measurement originates from each target that JIPDA makes in the data association process. To mitigate this issue a single-link hierarchical clustering method [17] is employed which merges any point within a specified distance threshold into a single cluster. The center of this cluster is then assumed to be the true detection.

4.2 Tracker outline

The tracker used in this work is a special case of the VIMMJIPDA [5] using only a single constant velocity (CV) model. In continuous-time this model is described by

$$\dot{\boldsymbol{x}} := \mathbf{A}\boldsymbol{x} + \mathbf{G}\boldsymbol{n} \tag{12}$$

with $\boldsymbol{x}$ as the current target state given by $\boldsymbol{x} = \begin{bmatrix} x_1 \; x_2 \; x_3 \; x_4 \end{bmatrix}^{\mathrm{T}}$ where x_1 and x_2 denote the north and east position while x_3 and x_4 are the corresponding velocities. The process noise $\boldsymbol{n}$ models target acceleration and is assumed to be white with diagonal covariance. The matrices $\mathbf{A}$ and $\mathbf{G}$ are defined as

$$\mathbf{A} := \begin{bmatrix} 0\,0\,1\,0 \\ 0\,0\,0\,1 \\ 0\,0\,0\,0 \\ 0\,0\,0\,0 \end{bmatrix} \qquad \mathbf{G} := \begin{bmatrix} 0\,0 \\ 0\,0 \\ 1\,0 \\ 0\,1 \end{bmatrix} \tag{13}$$

which in discrete form result in the model

$$\boldsymbol{x}_k := \mathbf{F}\boldsymbol{x}_{k-1} + v_k \qquad v_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k) \tag{14}$$

where $\mathbf{F}$ is the discrete state transition matrix and v_k the discretized process noise.

For each return the lidar measures the range of the object using time-of-flight as well as the direction of the return signal. By discarding height information this results in a sensor model using polar coordinates given by

$$f_z(\boldsymbol{x}_k) := \begin{bmatrix} \sqrt{x_1^2 + x_2^2} \\ \arctan(x_2/x_1) \end{bmatrix} + w_k \qquad w_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}) \tag{15}$$

where f_z is the measurement function and w_k the sensor noise described by the covariance matrix $\mathbf{R}$. Due to non-linearities in the measurement function, this would usually require an extended Kalman filter (EKF). However, by projecting the measurements into Cartesian coordinates [17] we can use the linear measurement model

$$f_z(\boldsymbol{x}_k) = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + w_k \qquad w_k \sim \mathcal{N}(\mathbf{0}, \mathbf{J}\mathbf{R}\mathbf{J}^{\mathrm{T}}) \tag{16}$$

where $\mathbf{J}$ is the Jacobian of the polar to Cartesian conversion and $\mathbf{R}$ the measurement noise in polar coordinates.

5 Datasets

The datasets used in this paper comes from experiments done in [15] where an autonomous ferry named milliAmpere was used as a research platform to record sensor data [4]. From this a synthetically reconstructed dataset was also made using the Gemini platform [14]. In this section we go through a brief description of the datasets from this study.

5.1 Setup

MilliAmpere was setup as *ownship* consisting of multiple exteroceptive sensors such as electro-optical and infrared cameras as well as radar and lidar. In addition, the ferry used a highly accurate navigation system based on Real-Time Kinematic GPS with sub-metre level position accuracy. To generate scenarios of multi-target interest, multiple targets equipped with GPS sensors for ground truth recording were used. Target 1, Havfruen, was used as a medium sized leisure craft capable of high speeds and rapid maneuvers. Target 2, Finn, functioned as a small leisure craft slower and less maneuverable than Havfruen. Target boats and ownship can be seen in Figure 3b.

5.2 Ground Truth Recording

Positional data of target vessels and ownship were recorded using different receivers as described in Table 2. Each vessel had 2 receivers in order to validate position, increase ground truth accuracy, and give a heading estimate later used by Gemini to generate correct ship orientations.

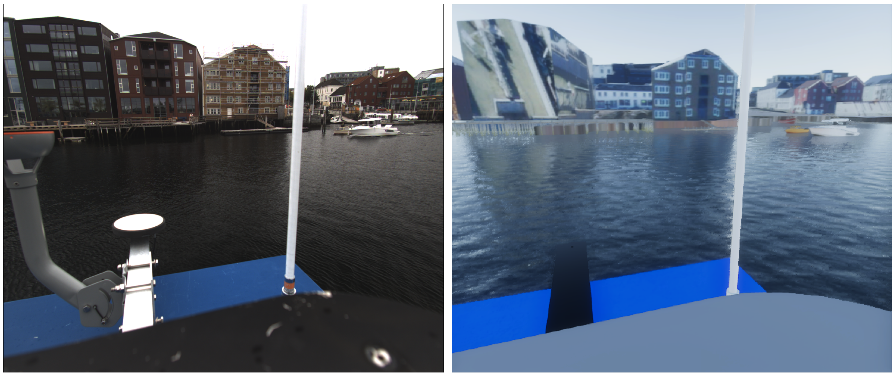


Fig. 2: Images from the visual analysis of the synthetic reconstruction done in [15]. Geometric and positional reconstruction of targets is intact, but discrepancies of the city model can be seen from e.g the missing red building.

.

Table 2: GNSS recievers.

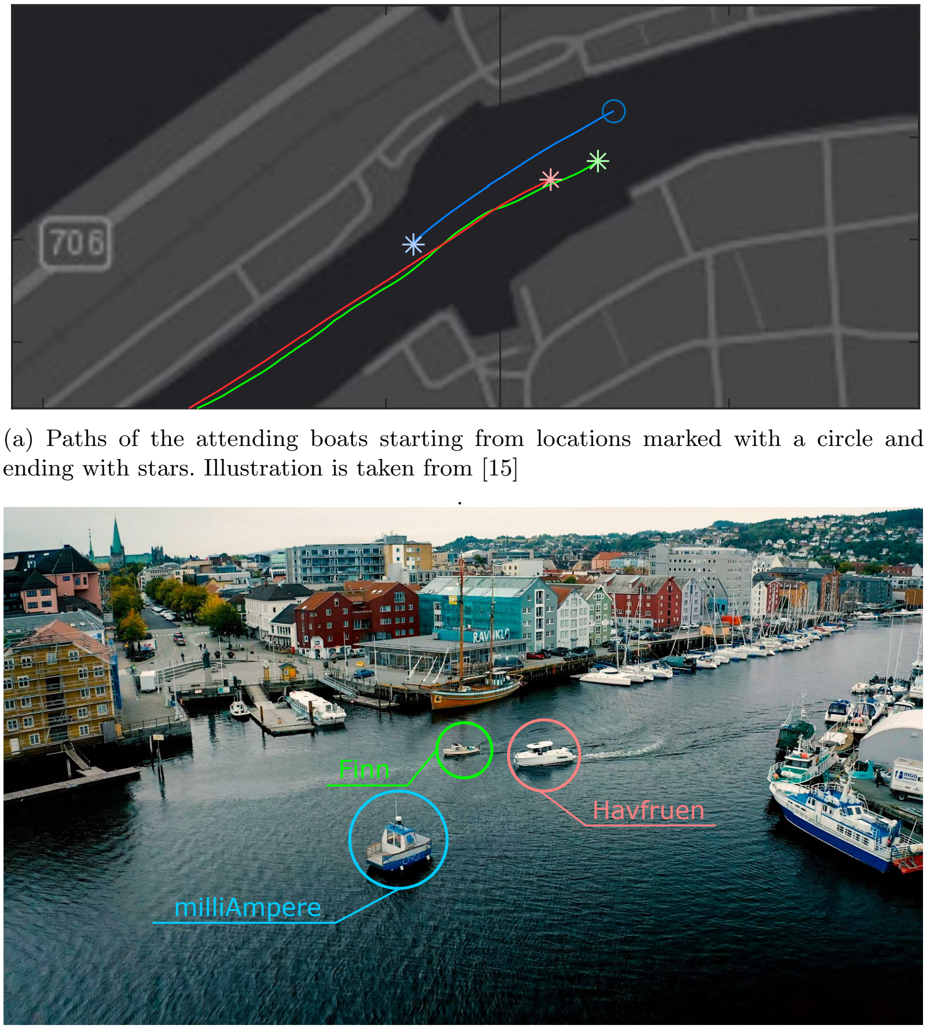
Ship	GNSS Receiver	Accuracy
milliAmpere	Hemisphere Vector VS330	1-10 cm
Havfruen	ZED-F9P	1 cm
Finn	Garmin eTrex 10	1 m

5.3 Scenarios

In the original study, a total of 9 scenario recordings was created. We chose to focus on scenario 8 with the target vessels following each other (Figure 3a). A drone footage of the scenario can be seen in Figure 3b that was recorded at Ravnkloa in the city of Trondheim.

5.4 Lidar

The experiment used a Velodyne VLP-16 puck that was later reconstructed synthetically using Gemini's lidar sensor [14] with improved beam modelling using among others a spherical projection filter [15, p.40]. To get suitable reconstruction of the real lidar data, 3D models of Trondheim city and participating target boats where used in conjunction with recorded ground truth data. The original data contained both camera and lidar data, while our analysis choose to use lidar only for simplicity. Instead the camera data was used as a visual confirmation to see the synthetic reconstruction besides real images from the ownship perspective (Figure 2).



(a) Paths of the attending boats starting from locations marked with a circle and ending with stars. Illustration is taken from [15]

.

(b) Scenario setup in the operating environment with the image facing south. Photo: Mikael Sætereid / Fosen innovasjon

Fig. 3: Illustrations of scenario 8. The attending boats Havfruen, Finn and milliAmpere (ownship) are coloured as red, green and blue respectively in each illustration.

6 Evaluation

Our intention is to study effects that contribute to the Hellinger metrics derived in Section 3, and what relations they have to each other.

Previous analysis of the dataset showed the ground truth for the ownship velocity to be noisy [15, p. 92]. Moreover, getting good velocity estimates for VIMMJIPDA trackers have in addition proven itself to be difficult [5], especially for targets with large extensions. The synthetic data also have discrepancies due to incomplete 3D models as seen earlier in Figure 2. To lessen the influence of these known effects, we choose to run the Hellinger metrics on tracks in near proximity of the target vessels and disregard the velocity estimates from the tracker.

We have chosen to study the remaining effects by comparing the metrics in context of how tracks overlap in position when generated by real and synthetic data (Figure 4). Each track are here represented as a covariance ellipse based of a 95% confidence interval from the Gaussian distribution it represents (1).

6.1 Track Association

For Hellinger and Bernoulli Hellinger a validation gate with radius 5m centered at ground truth is used for track association (Figure 4). Tracks outside the gate are discriminated, while if more than one track from a dataset is present in the gate, the closest estimate to ground truth is chosen. For the MB-Hellinger, no association method is required since each track is compared to each other weighted by their existence probability. As a result, from Table 3 we have a low MB-Hellinger distance in the first case while for Hellinger and Bernoulli Hellinger the distances are high at the same time instances since there's no pair of tracks inside their gates. Furthermore, the MB-Hellinger is always defined since it does not need validation gates that risks being empty as happens with Finn in two cases.

6.2 False Tracks

False tracks from the datasets can be seen in Figure 4 as ellipses without a real or synthetic counterpart. Due to their high existence probabilities we get a large MB-Hellinger distance. By manually downweighting the existence probabilities of these tracks, we see in Figure 4c a large effect when comparing the normal and weighted MB-Hellinger.

6.3 Bernoulli Hellinger a Special Case of MB-Hellinger

In (11) we showed that the Hellinger is a special case of the Bernoulli Hellinger. We also stated in section 2.2 to the Bernoulli Hellinger being a special case of MB-Hellinger. In Figure 4c we have a situation with Havfruen where we can see tendencies of this relationship. Also the succeeding dip for Finn in the same figure can be seen in the MB-Hellinger metric as well. It is worth noting that in these occasions the MB-Hellinger is always bigger than Bernoulli, which can be explained by MB evaluating all tracks instead of specific tracks as with the other metrics. This shows how difficult it is for MB-Hellinger to be equal to Bernoulli Hellinger in comparison to Hellinger being equal to Bernoulli Hellinger as seen in the metric plots for either Havfruen or Finn in Figure 4.

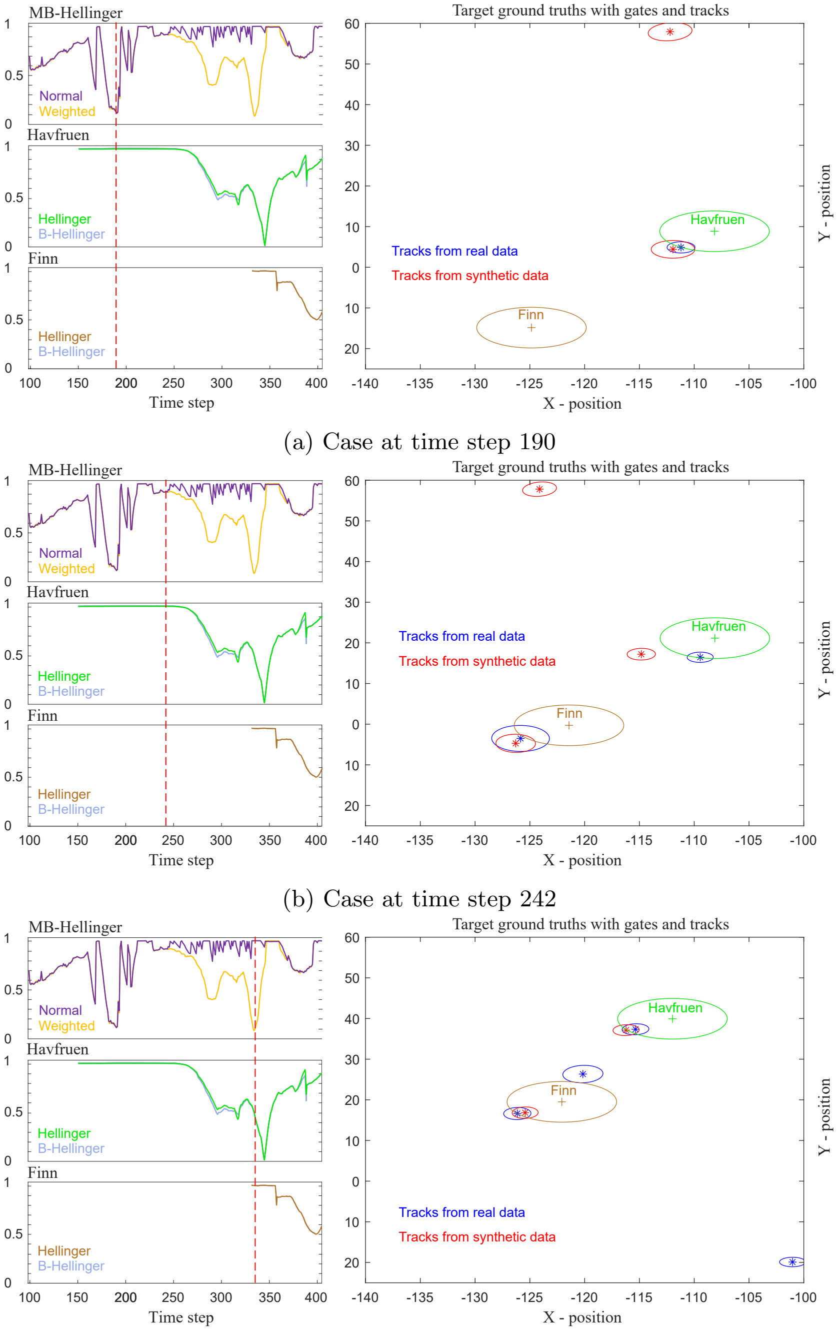
Table 3: Metric results for normal (N) and weighted (W) MB-Hellinger, Bernoulli-Hellinger (B-H) and Hellinger (H) from cases shown in Figure 4
.

	MB-Hellinger		Havfruen		Finn	
Case	N	W	H	B-H	H	B-H
a	0.11	0.11	1.00	1.00	N/A	N/A
b	0.92	0.92	1.00	1.00	N/A	N/A
c	1.00	0.10	0.44	0.44	1.00	1.00

7 Discussion

In this paper we have been focusing on analysing results from a real and synthetic lidar sensor. It is, however, worth pointing out that because of how the Bayesian filter works, the method presented here can be used for all sensor types given there is a detection model (e.g as in section 4.1), and models on how to handle the detections in the filter (e.g as in section 4.2). This have been previously demonstrated in [15] where the impact of individual sensors (such as lidars and cameras) on the tracker could be quantified using the metric.

A big difference between single versus multi distribution metrics is the way track association between the datasets are handled. From Figure 4a we can see that increasing the validation gates slightly would have allowed Hellinger and Bernoulli-Hellinger to give a measure on Havfruen. This measure would likely have been closer to that of MB-Hellinger rather than 1. Similarly, the gates also causes trouble in several cases for Finn, where the metric becomes undefined when the gates are empty (Table 3). A solution could be to say empty gates gives a metric value of 0, but given that there might be tracks just outside the gate as we see in Figure 4b, moreover that a low metric value would bias the results to seem better from the lack of track data, the proper way of handling an empty gate becomes questionable. In comparison to dealing with gates, this is a task easier handled by MB's checking and weighting each tracks permutation. However, if downgrading of existence probabilities for false tracks is necessary, one would need to argue why this tuning is needed for comparing datasets. One reason could be if the tracker is overconfident on false tracks stemming from differences such as water reflections. This would be both computationally



(a) Case at time step 190

(b) Case at time step 242

(c) Case at time step 334

Fig. 4: Cases of interest. Left: Distances for single and multi-target Hellinger metrics. Right: covariance plot for tracks with ground truth and validation gates.

demanding to simulate and to properly reconstruct, while likely being a non crucial difference between synthetic and real data for autonomy purposes.

The close similarity of Bernoulli-Hellinger and Hellinger for both Finn and Havfruen in Figure 4 and the fact that this only happens when the existence probabilities in both datasets are close to 1 (section 3.2), shows that the JIPDA is overconfident on tracks close to the targets. Furthermore, the false tracks in Figure 4c contribute 0.9 to the metric when looking at the difference between the normal and weighted MB-Hellinger in Table 3. This shows how much false tracks impacts the MB-Hellinger metric when dealing with an overconfident JIPDA. If we argue that the situation depicted in Figure 4c should have resulted in a lower metric value, the Hellinger metric might be to strict in comparison to other metric candidates suited for the Csiszár Information Functional.

False tracks is not the only discrepancy we have between the datasets. Even after removing these and the presumed sub-optimal velocity estimates, there is still fairly high metric values with huge spread in range over time. These differences could range from transformation errors due to sensor mountings to environmental reproduction discrepancies as seen in Figure 2 that have a substantial enough effect on the tracker. More elaborate visualisation techniques are needed to see metrics, sensor data and the situational awareness picture layered on top of each other or being interchanged to make further analysis on this.

What might be of benefit in this regard is the ability to do single target analysis such as Hellinger and Bernoulli Hellinger in contrary to the MB-Hellinger. For judging sensor fidelity where a complete environment reconstruction is not possible, studying single targets with proper ground truth and 3D models might be easier to do and quantify. MB-Hellinger on the other hand takes a more global approach of measuring everything found in a specified area, potentially including unfortunate discrepancies as seen in the missing building in Figure 2. The land filtering done as a preprocessing step in the JIPDA pipeline can accommodate for portions of this, but in uncontrolled environments where autonomy operates, even a bird which have not been accommodated for could show up as a false track not seen in the simulated dataset. If the tracker is over confident on these tracks, the result will be high MB-Hellinger distances which is the tendency seen in Figure 4. On the other hand, if the goal of the metric is to measure the complete reprodusability of an experiment including that of false tracks, this may still be of benefit. Otherwise for purposes concerning sensor modeling, MB-Hellinger would need a better method for discriminating the unintended tracks in the environment.

8 Conclusion

In this paper we have shown the use of single-target and multi-target Hellinger metrics for quantifying the performance difference of a multi-target tracker when subjected to real and synthetically reconstructed data. We have demonstrated how the Hellinger distance can be used in various ways to judge single target as well as multi-target comparisons, their relationship to each other in addition to their various pros and cons in analysis work. From this the paper have contributed with a method of evaluating the MB-Hellinger by means of importance sampling. In addition we have presented results from a use case where the metric is used for comparing real and simulated data created with respect to validating SITAW systems. Further work includes more sophisticated visualisation tools to be able to explain and analyse the remaining Hellinger distance obtained in the results. Due to the close dependency on a tracker for establishing the metrics in the first place, an interesting study would also be to see how well the metrics would perform as an alternative to existing validation metrics such as COSPA and OSPA. In this work exploring other distances such as the Kullback-Leibler in Csiszár's information functionals and doing a sensitivity analysis of existence would be in place.

Acknowledgment

This work was supported in part by the Research Council of Norway through Projects 223254, 331921 and 309230, and by NTNU through the Autoferry project.

References

1. Abou-Moustafa, K.T., Torre, F.D.L., Ferrie, F.P.: Designing a Metric for the Difference between Gaussian Densities. In: Brain, Body and Machine. AINSC, vol. 83, pp. 57–70 (2010). https://doi.org/10.1007/978-3-642-16259-6_5
2. Acuna, D., Zhang, G., Law, M.T., Fidler, S.: f-Domain Adversarial Learning: Theory and Algorithms. In: Proceedings of the 38th International Conference on Machine Learning. PMLR, vol. 139, pp. 66–75 (2021)
3. Bar-Shalom, Y., Tse, E.: Tracking in a Cluttered Environment with Probabilistic Data Association. Automatica **11**(5) (1975). https://doi.org/10.1016/0005-1098(75)90021-7
4. Brekke, E.F., Eide, E., Eriksen, B.O.H., et al: milliAmpere: An Autonomous Ferry Prototype. Journal of Physics: Conference Series **2311**(1) (2022). https://doi.org/10.1088/1742-6596/2311/1/012029
5. Brekke, E.F., Hem, A.G., Tokle, L.C.N.: Multitarget Tracking With Multiple Models and Visibility: Derivation and Verification on Maritime Radar Data. IEEE Journal of Oceanic Engineering **46**(4) (2021). https://doi.org/10.1109/JOE.2021.3081174
6. Deitke, M., Han, W., Herrasti, A., et al: RoboTHOR: An Open Simulation-to-Real Embodied AI Platform. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3161–3171 (2020). https://doi.org/10.1109/CVPR42600.2020.00323
7. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., et al: CARLA: An open urban driving simulator. In: Proceedings of the 1st Annual Conference on Robot Learning. PMLR, vol. 78, pp. 1–16 (2017)
8. Mahler, R.P.S.: Advances in Statistical Multisource-Multitarget Information Fusion. Artech House (2014)

9. Musicki, D., Evans, R., Stankovic, S.: Integrated Probabilistic Data Association (IPDA). In: The 31st IEEE Conference on Decision and Control. vol. 4, pp. 3796–3798 (1992). https://doi.org/10.1109/CDC.1992.370951
10. Musicki, D., Evans, R.: Joint Integrated Probabilistic Data Association: JIPDA. IEEE transactions on Aerospace and Electronic Systems **40**(3) (2004). https://doi.org/10.1109/TAES.2004.1337482
11. Richter, S.R., AlHaija, H.A., Koltun, V.: Enhancing Photorealism Enhancement. arXiv (2021). https://doi.org/10.48550/ARXIV.2105.04619, preprint
12. Shah, S., Dey, D., Lovett, C., et al: AirSim: High-fidelity visual and physical simulation for autonomous vehicles. In: Field and Service Robotics. SPAR, vol. 1, pp. 621–635 (2018). https://doi.org/10.1007/978-3-319-67361-5_40
13. Tobin, J., Fong, R., Ray, A., et al: Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 23–30 (2017). https://doi.org/10.1109/IROS.2017.8202133
14. Vasstein, K., Brekke, E.F., Mester, R., et al: Autoferry Gemini: A Real-Time Simulation Platform for Electromagnetic Radiation Sensors on Autonomous Ships. IOP Conference Series: Materials Science and Engineering **929** (2020). https://doi.org/10.1088/1757-899x/929/1/012032
15. Vasstein, K.: A High Fidelity Digital Twin Framework for testing Exteroceptive Perception of Autonomous Vessels. Master's thesis, NTNU (2021), https://hdl.handle.net/11250/2781031
16. Williams, J.: Marginal multi-Bernoulli filters: RFS derivation of MHT, JIPDA, and association-based MeMBer. IEEE Transactions on Aerospace and Electronic Systems **51**(3) (2015). https://doi.org/10.1109/TAES.2015.130550
17. Wilthil, E.F., Flåten, A.L., Brekke, E.F.: A target tracking system for ASV collision avoidance based on the PDAF. In: Sensing and Control for Autonomous Vehicles, LNCIS, vol. 474, pp. 269–288. Springer (2017). https://doi.org/10.1007/978-3-319-55372-6_13
18. Zajic, T., Mahler, R.P.S.: Practical information-based data fusion performance evaluation. In: Proceedings of Signal Processing, Sensor Fusion, and Target Recognition VIII. vol. 3720, pp. 92–103 (1999). https://doi.org/10.1117/12.357148
19. Zhu, J.Y., Park, T., Isola, P., et al: Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. In: IEEE International Conference on Computer Vision. pp. 2242–2251 (2017). https://doi.org/10.1109/ICCV.2017.244