December 2022 Volume 17 Number 2 ISSN 1557-6418

JAIF 22

Journal of Advances in Information Fusion

A semi-annual archival publication of the International Society of Information Fusion

www.isif.org

ISIF

INTERNATIONAL SOCIETY OF INFORMATION FUSION

The International Society of Information Fusion (ISIF) is the premier professional society and global information resource for multidisciplinary approaches for theoretical and applied INFORMATION FUSION technologies. Technical areas of interest include target tracking, detection theory, applications for information fusion methods, image fusion, fusion systems architectures and management issues, classification, learning, data mining, Bayesian and reasoning methods.

Manuscripts are submitted at http://jaif.msubmit.net. If in doubt about the proper editorial area of a contribution, submit it under the unknown area.

Journal of Advances in Information Fusion (ISSN 1557-6418) is published semi-annually by the International Society of Information Fusion. The responsibility for the contents rests upon the authors and not upon ISIF, the Society, or its members. ISIF is a California Nonprofit Public Benefit Corporation at P.O. Box 4631, Mountain View, California 94040.

Modeling Spatial Uncertainty for the iPad Pro Depth Sensor

ANTONIO ZEA
UWE D. HANEBECK

Depth sensors, once exclusively found in research laboratories, are quickly becoming ubiquitous in the mass market. After Apple's introduction of the iPad Pro 2020 with an integrated light detection and ranging (LIDAR) sensor, now even tablets and smartphones are capable of obtaining accurate 3-D information from their environments. This, in turn, increases the reach of applications from technical fields, such as SLAM, object tracking, and object classification, which can now be downloaded on millions of hand-held devices with a couple of taps. This motivates an analysis of the capabilities, strengths, and weaknesses of these depth streams. In this paper, we present a study of the spatial uncertainties of the iPad Pro 2021 depth sensor. First, we describe the hardware used by the device, and provide an overview of the machine learning algorithm that fuses information from the LIDAR sensor with color data to produce a depth image. Then, we analyze the accuracy and precision of the measured depth values, while giving attention to the resulting temporal and spatial correlations. Another important topic of discussion are the tradeoffs involved in the extrapolations that the depth system implements, such as how curvatures change at different distances. In order to establish a reference baseline, we also compare the obtained results to another widely known time-of-flight sensor, the Microsoft Kinect 2.

Manuscript received January 18, 2022; revised August 10, 2022; released for publication March 20, 2023

Antonio Zea and Uwe D. Hanebeck are with the Intelligent Sensor-Actuator-Systems Laboratory (ISAS), Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology (KIT), 76131 Germany (e-mail: antonio.zea@kit.edu, uwe.hanebeck@ieee.org).

Refereeing of this contribution was handled by Florian Meyer.

This work was supported by the German Federal Ministry of Education and Research through ROBDEKON Project under Grant 13N14675.

1557-6418/22/$17.00 © 2022 JAIF

I. INTRODUCTION

Depth sensors have been used for several decades in a wide variety of fields, ranging from robotics to computer-aided design (CAD), medicine, entertainment, and even the arts. During this time, multiple depth sensing technologies have been developed based on different operating principles. For example, visual data, such as RGB or grayscale images, can be processed to determine disparities at given points, which in turn can be used to reconstruct depth information. Recent advances in machine learning (ML) [7], [10], [11] can even extrapolate depth from a static image by filling in missing information from previously trained scenes. Light detection and ranging (LIDAR) sensors measure depth by projecting a (usually rotating) laser pulse onto a scene and calculating the time it takes for the pulse to return. In time-of-flight (ToF) cameras, a specialization of this technology, the laser pulse is split into a dense array of thousands of points, measuring an entire scene in a single scan and achieving a resolution and frame rate comparable to small RGB cameras.

Until about a decade ago, depth sensors in general were out of reach of the mass market due to their price. This changed significantly when Microsoft introduced the Kinect in 2010, an affordable depth camera (150 Euro) based on structured light. Originally designed as a body tracker for the Xbox 360, it was quickly adopted everywhere from robotics [19] to metrology [22] and therapeutics [16]. Since then, depth sensors have become ubiquitous in the research and hobbyist communities, with newer generations increasing accuracy and robustness while reducing their size and price. However, they have generally remained separate stand-alone devices. This stands in contrast to RGB cameras, most of which are now integrated into PCs, laptops, and mobile devices.

Among the first commercial attempts to integrate depth sensors into mobile devices were Lenovo's Phab 2 PRO in conjunction with Google's Tango platform [20] in 2016. Intended applications included immersive augmented reality (AR) experiences, scene reconstruction, and indoor tracking. However, there were few apps capable of exploiting these capabilities, leading to low interest from consumers. In turn, Tango was discontinued in 2017 and replaced with ARCore [13], which extracts depth information from RGB images and ML postprocessing. Since then, other smartphone manufacturers have made integration attempts, such as the Samsung Galaxy S20 with a ToF sensor, but it was discontinued for the release of the S22. Microsoft also integrated ToF sensors into its Hololens devices for AR (both first- and second-gen), but since 2019, they have made no concrete announcements about a third-gen device. These ebbs and flows of depth sensor integration are a consequence of a developer chicken-and-egg problem: if there are no apps, users will not buy the devices, but if users are not interested, then devel-

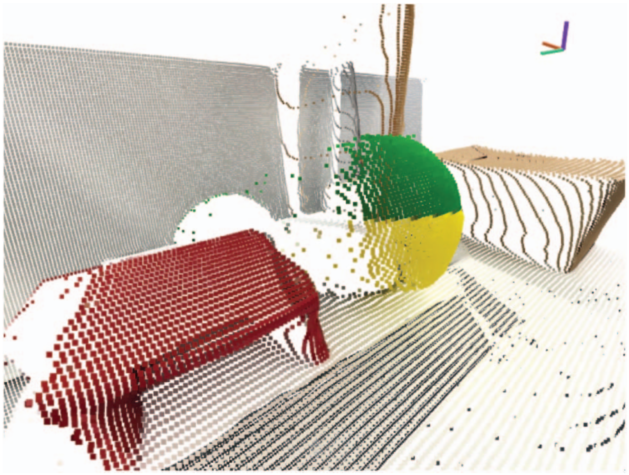


Fig. 1. Example of reconstructed point cloud for a table scene with objects, captured by an iPad Pro 2021 tablet.

opers have no reason to implement apps in the first place.

However, this situation is quickly changing with the introduction of platforms, such as Apple's ARKit [3], and the entrance of new competitors, such as Magic Leap and Facebook (now Meta). The newly introduced depth sensors of Apple are particularly noteworthy. In 2017, they announced a depth sensing technology on the iPhone X called "TrueDepth," which uses an infrared dot pattern similar to the first generation Kinect and has a range of up to 40 cm. Later, in 2020, they released a ToF sensor on the back of the iPad Pro with range of up to 5 m (see Fig. 1 for an example scene). Currently, all of Apple's mid and high-tier smartphones and tablets come with a depth sensor, and given the significant market share of their mobile devices, it is likely that their depth sensing technology will be the most widely used among nontechnical users in the near future. In this brief time, the iPad and iPhone LIDARs have already found versatile applications outside of the intended target of AR, ranging from forestry [12], [29], [31] and architecture [27] to heritage documentation [23] and geology [18], and even veterinary medicine [21]. This motivates an in-depth analysis of these sensors, in order to determine their usability in fields, such as localization and tracking. Similar depth sensors, such as the Microsoft Kinect (first and second-gen), have been extensively studied in literature [5], [8], [26], [32]. The TrueDepth system has already received some attention, for example, in [4] and [30], where the iPad is contrasted with an industrial scanner. The measurement accuracy of the iPhone 12 LIDAR [17] and the iPad Pro 2020 LIDAR [23], [30] has also been evaluated, but in the context of 3-D scanning and static reconstruction. However, we have not been able to find works that deal with quantitative models of spatial uncertainties for the iPad LIDAR.

The rest of this paper is organized as follows. Section II contains a description of the depth sensor, including the hardware and the streams provided by the API. Section III introduces a quantitative analysis of the measurements provided by the depth streams and their uncertainties. Finally, Section IV concludes this article. Throughout the paper, we will compare the iPad depth sensor to the well-known Kinect 2 device, which will serve as a baseline for its capabilities. However, we emphasize that these comparisons are merely illustrative, as both devices have different capabilities and are not aimed at the same range of applications.

II. IPAD DEPTH SYSTEM

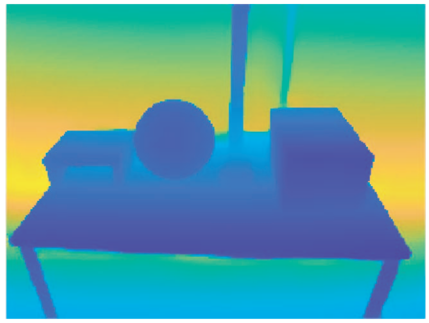
The iPad depth sensor was introduced with the iPad Pro 2020 [2], the first Apple device to use a ToF sensor. It works by fusing depth and color streams together using ML algorithms. Its main application is in ARKit, Apple's AR platform, where it is employed for depth occlusion, scene understanding, and the detection, segmentation, and tracking of objects and humans. Fig. 2 shows an example of the data streams provided by the device, which when converted into a point cloud produced the image seen in Fig. 1.

The LIDAR sensor is located on the back (or rear) of the tablet, i.e., the side pointing away from the user (see Fig. 3). The hardware appears to be identical in both the 11 and the 12.9 in variants of the iPad Pro 2020 and the iPad Pro 2021, and a related study [17] found no difference with the iPhone 12. Also note that the front of the device (the display side) provides another depth system called "TrueDepth," based on structured light. Its main use is face recognition for authentication (FaceID) and face tracking, employed, for example, in Snapchat filters and "lenses." Both depth systems also differ in their operating range. While TrueDepth works best at a distance of at most 40–50 cm [30], the LIDAR sensor can measure walls up to 5 m away with moderate accuracy. This distinction also affects where they can be used. For example, while detecting small objects on a table is better suited for the TrueDepth system, a localization application in a large room would prefer the LIDAR data instead. Note that the TrueDepth system will *not* be considered in this work.

A. Sensors and Data Streams

We start by introducing the sensors on the back of the tablet, as shown in Fig. 3. On the top left is a wide-angle RGB camera, which from our experiments does not appear to be used in the depth system. On the top right is the standard RGB camera, with a smaller field of view but higher resolution. On the bottom left is the infrared flood illuminator, which ensures that the scene has an appropriate amount of light for the LIDAR system. On the bottom right is the flashlight, also not used by the depth system. Finally, the ToF sensor at the center consists of two submodules [28]. On the one hand, a vertical-cavity surface-emitting laser (VCSEL) diode is in charge of emitting a laser pulse which is split by


(a) Color image


(b) Depth image (in colormap)

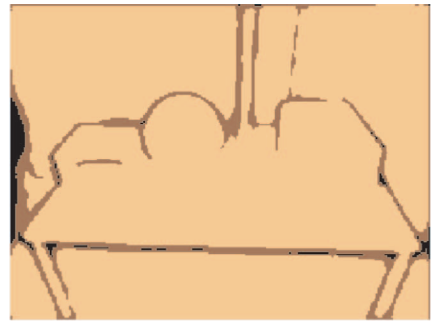

(c) Confidence image (in colormap)

Fig. 2. Example capture of the scene from Fig. 1 showing the three video streams of the iPad depth system: the color image (1920 × 1440 px, RGB), the depth image (256 × 192 px, 32-bit float array), and the confidence image with 256 × 192 pixels, 1 byte/px, and three possible discrete values: low (black), middle (brown), and high (orange) confidence. (a) Color image. (b) Depth image (in colormap). (c) Confidence image (in colormap).

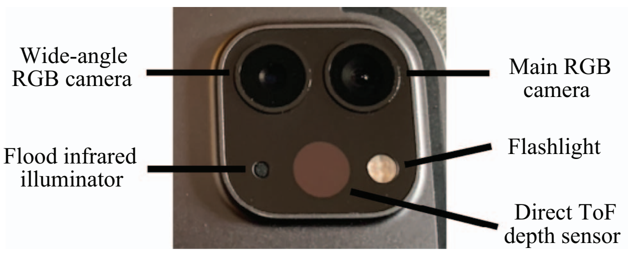


Fig. 3. Sensors on the back of the iPad Pro 2021. Width and height of the panel are around 27 mm.

a lens into 3 × 3 blocks with 8 × 8 dots each [1], [17], which are then projected onto the scene. A sketch of the pattern can be seen in the left-hand side of Fig. 4. On the other hand, a CMOS sensor captures the reflected light and calculates the distance between the device and each of those dots. Note that this suggests that each depth frame is interpolated from only $24 \times 24 = 9 \times 64 = 576$ measurements [1].

The depth data from these pulses are not directly available from the software library. Instead, the ARKit platform processes this information internally and fuses it with the color stream to produce a depth estimate. The result is a stream of three images at 60 frames/s (Fig. 4, right-hand side): a color image, a "scene" depth image, and a confidence image (see Fig. 2). For the sake of completeness, we note that the depth API also provides other streams, such as a real-time mesh reconstruction and a point cloud of RGB feature points, which will not be considered in this work. The values in this paper were captured using ARKit 5 and iOS 15.2.

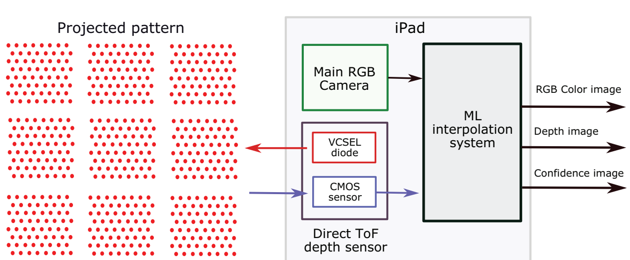
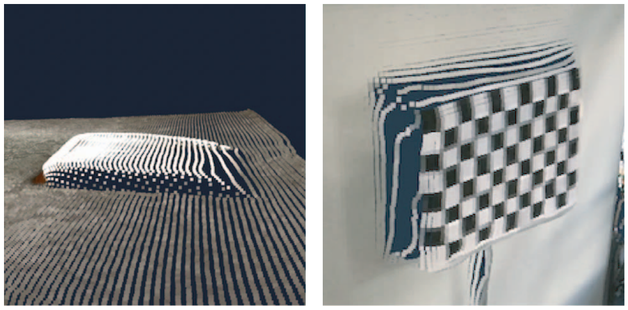


Fig. 4. Sketch of the depth capture process: a pattern of 576 dots is projected onto the scene, their distances are measured by the depth sensor, and the result is fused with the data from the RGB camera. The result are three streams: the color image, the depth image, and the confidence image.

The color image [see Fig. 2(a)] is an uncompressed packed RGB stream with a resolution of 1920 × 1440 pixels (px) and 24 bits/px. The original YUV image is also accessible if desired. The depth image [see Fig. 2(b)] has a resolution of 256 × 192 px, and each pixel contains a 32-b float describing the depth in meters at that position. The depth image is already registered to the color image and has the same aspect ratio of 4 to 3. The field of view of both images is about 60° horizontal and 48° vertical, with slight variations between devices. Finally, ARKit also provides a confidence map [see Fig. 2(c)] which determines how accurate the depth value of each pixel is. It has the same resolution as the depth image, but each pixel has an 8-bit integer value, which can be either 0 (low), 1 (medium), or 2 (high). Unlike sensors, such as the Microsoft Kinect, an invalid measurement is not encoded with a depth value of 0. Instead, the corresponding confidence is set to 0, and the depth is extrapolated based on surrounding depth values and semantic cues from the color image. In Fig. 2, a confidence of 0 can be seen around the edges of the table, or at the image borders. However, these gaps are not evident when looking at the depth image on its own.

The three images in Fig. 2 correspond to the point cloud that was shown in Fig. 1. Here, the large amount of "fringe points" (also known as "flying pixels") at the borders are clearly visible, as a result of the relatively low LIDAR resolution. Looking at the brown cardboard box at the back, it is also clear that planar surfaces are not necessarily shown flat, and that 90^o corners are not usually preserved. More interestingly, we observe that the measurement noise is strongly spatially correlated but not temporally correlated. In other words, unlike depth sensors, such as the Microsoft Kinect 2, where each measurement moves back and forth independently from its neighbors, here we observe entire sur-



(a) Box with height 2 cm (b) Chessboard at 20 cm from a wall

Fig. 5. Illustration of how the iPad observes objects with steep depth changes. (a) Box with height 2 cm. (b) Chessboard at 20 cm from a wall.

faces appearing to move and deform "coherently" each frame, but in slightly different ways. This tendency can be appreciated, for example, in the fringe points of the tall brown cylinder, which form different curved lines each frame. These correlations are introduced by the interpolation system, which will be described in Section II-B.

B. ML Interpolation System

The ML interpolation system is in charge of filling the gaps between the sparse depth measurements by fusing data from multiple streams, in particular from the LIDAR and RGB sensors. A sketch of its workings can be found in the patent description [33]. Unfortunately, as of iOS 15, there is no way to turn it OFF and obtain the raw data. However, given that it has a significant effect on the provided measurements, it is of interest to describe qualitatively how it works and what it does.

Generally speaking, the iPad depth system acts by aggregating measurements into coherent surfaces and "smoothing" out sudden changes in depth. This can be seen in Fig. 5(a), where the back of the white box, being observed from 1 m away, merges into the floor. The capability of the iPad to discern depth discontinuities is reduced by the sensor distance, as can be seen in Fig. 5(b) with the sensor being 4 m away, where the chessboard "melts" into the wall. These smoothed out measurements can usually be identified by their confidence level of 1 (medium) or 0 (low), as shown in Fig. 2(c) in dark brown and black, allowing them to be filtered out.

As mentioned before, pixel positions with no valid depth are denoted with a confidence level of 0. Invalid measurements can happen for several reasons, such as a reflective or bright material, a drastic change in depth, or the depth being outside of the operative range. Very narrow objects will also fail to be detected, especially if their size is less than of about 6% of the image (i.e., 15 px for the depth image). This value is probably related to the distance between LIDAR dots, i.e., 256 px/row or 24 dots/row $\approx$ 11 px. Similarly, detection will also fail if the depth changes quickly at the image edges [see Fig. 2(c)]. For these regions, the depth image will not show invalid values. Instead, the ML interpolation system will fill in the gaps by "guessing" which depths belong there based on data from the color image.

As the ML interpolation combines data from both the color and depth images, it is of interest to see how they work when one data stream is unavailable, for example, by covering one camera with tape or a piece of paper. When the color stream is absent, the resulting depth image appears extremely blurred, lacking sharp corners and with significant sections of the confidence image with values of middle or low. When the depth sensor is covered, the entire confidence image has a value of low, and the depth inference from color is applied to the entire image. This can produce interesting results, such as seen in Fig. 6. The setup is an iPad tablet 20 cm away from a flat monitor that is showing an image of a rendered cube on a plane. Fig. 6(a) presents the RGB capture from the color camera. Fig. 6(b) illustrates the resulting point cloud with the depth and color sensors active. However, if we cover the color camera, the depth inference generates a scene with a cube at a distance of 2 m [see Fig. 6(c)]. As with the fringe points, these reconstructed depths possess spatial correlations but lack temporal correlation, and thus, will change drastically between frames. Note that these reconstructions do not only appear if the whole depth sensor is covered. For example, a shiny, reflective object somewhere in a scene (such as a laminated picture) can produce the same effect.

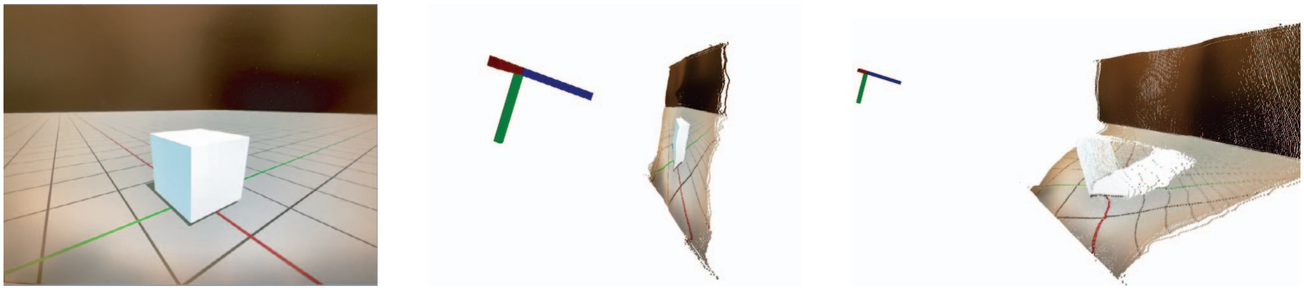
III. MODELING SPATIAL UNCERTAINTY

In this section, we will present a quantitative analysis of the uncertainties in the iPad depth system. First, we start by analyzing the depth discretization, which tells us the range of values that the system can provide. Then, we will measure the measurement bias, that is, the difference between the measured and the real depths. After this, we will focus on stochastic uncertainties and establish a measure for the correlations (in time and space) between measurements. Finally, we will present an analysis of how the measured curvature degrades as a function of distance.

A. Discretization

An important aspect to determine the quality of the depth stream is to see which depth values can be represented in the first place. The Kinect 1, for example, can only produce 2048 depth values spread out over the operating range of 0 to 8 m. The Kinect 2 has a much higher resolution, but as the depths are provided in millimeters as 16-b unsigned integers the distance between measurements are necessarily multiples of 1 mm. In contrast to these sensors, the resolution of the iPad depth system depends on the depth range.

Fig. 7 shows the quantization (also known as discretization) of the depth values, i.e., the distance between one value and the next, in function of the depth. This dis-



(a) Monitor showing a virtual scene (b) Point cloud with color and depth active (c) Inferred point cloud from color image

Fig. 6. Example of ML depth inference where the iPad observes a computer monitor from a closeup distance of 20 cm. When the depth sensor is covered, the iPad depth system takes the color image and extrapolates it into a new scene where the cube sits at a distance of 2 m. (a) Monitor showing a virtual scene. (b) Point cloud with color and depth active. (c) Inferred point cloud from color image.

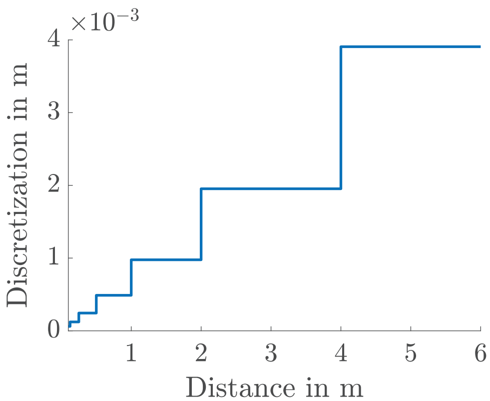


Fig. 7. Discretization of depth values, i.e., the distance between a value and the next possible one, depending on the distance.

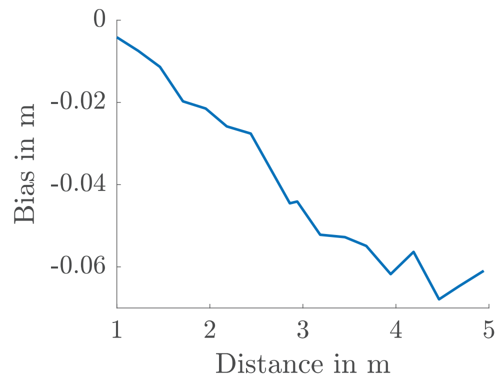


Fig. 8. Difference between the ground truth depths and the measured depths at difference distances. The negative bias means that the measured values are smaller than the ground truth.

cretization appears to be structured so that, between one power of two and the next (in meters), there are 1024 values. Thus, between 1 and 2 m we will see values at spaces of $1/1024$ m, or around 0.977 mm. However, between 2 m and 4 m, the space will be $1/512$ m instead, i.e., twice as large. Thus, the lower the resolution becomes, the larger the depth value is.

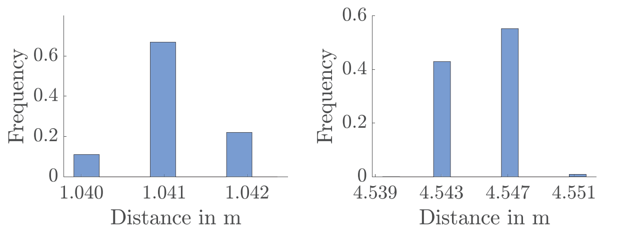
The range of possible values is difficult to measure. Any object farther than 5 m will have its measurements automatically marked as having confidence 0, and thus, the received depths will originate from the depth inference system and not from the LIDAR. Still, in our experiments, we have not observed a depth value above 8 m. Values below 10 cm will similarly be marked as low or medium confidence, yielding "reconstructed" depths that may have little relation to the actual physical distance.

B. Measurement Bias

As usual in real-life sensors, the measured depth values are not the true depths, as the process of capturing the data introduces errors, which depend on many factors, such as the pixel position, material, angle of reflection, temperature, and others. In order to keep the model simple, we can divide these errors into two additive components: a fixed offset (bias) and a zero-mean stochastic noise term. Both terms depend on the depth from which the measurement originated.

In this section, we will focus on estimating how the bias behaves at different distances. To achieve this, we captured 150 frames of a paper chessboard [70×49 cm, see Fig. 5(b)] at different distances, ranging from 1 to 5 m. The ground truth depth was obtained by detecting the chessboard in the color stream, estimating the board plane using MATLAB's implementation of the *solvePnP* algorithm, and finally calculating the depth that corresponded to the plane center. For the measured depth, we considered a square 5×5 px window around the center of the detected chessboard, and then calculated the average of all values for all frames. All of the considered points have a confidence of 2 (high). Fig. 8 shows the results.

It can be seen that the bias was always negative, that is, the iPad tells us that the object is closer than it really is. The bias also appears to increase linearly, but still remaining with 1% to 2% of the ground truth. However, this model stops being reliable at about 4 m, given the tendency of the chessboard surface to "melt" into the back wall at large distances. In these cases, the average depth can change significantly depending on where on the chessboard the window is located. After 5 m, the depth inference system kicks in, causing the obtained values to bear little relation to the real depths.



(a) Depth distribution at 1 m (b) Depth distribution at 4.5 m

Fig. 9. Representative probability distributions for nonfringe depths values at different distances. The gaps between values are due to the discretization. (a) Depth distribution at 1 m. (b) Depth distribution at 4.5 m.

C. Measurement Noise

In this section, we use the same data collected in Section III-B to analyze how the noise term of the measured depth behaves. In particular, we will focus on four aspects: the probability distribution of the noise, its variance depending on the depth, and the magnitude of spatial and temporal correlations.

Fig. 9 shows the distributions of nonfringe depths values captured at representative positions. These are discrete distributions, and the gaps between values correspond to the discretization described in Section III-A. At short distances, the support consists of two or three values, as shown in Fig. 9(a). For higher distances, the support becomes slightly wider, reaching up to four values in Fig. 9(b). In any case, the distributions are consistently unimodal and appear roughly symmetrical around the mean (prequantization), and thus, we suspect that approximating them as Gaussians in practical applications will not lead to much loss of information.

When dealing with estimators, it is also important to know the variance that corresponds to a given measurement, preferably without having to wait for additional measurements from the same position. Fig. 10 shows the variances of the measurements gathered in the chessboard dataset from Section III-A. In yellow, we observe the variance stemming from the discretization, assuming a uniform distribution that ranges between the previous and the next possible values. In blue, we see the sample variance gathered from 150 frames. In red, we see a best-fitting third degree polynomial with the form

$$\sigma_z^2 = 10^{-6} \cdot (0.07z^3 - 0.32z^2 + 0.64z - 0.02), \qquad (1)$$

which closely approximates this empiric variance and can be easily integrated into an estimator.

Given a depth value z at the pixel position $[u, v]^T$, i.e., at column u and row v in the depth image, it is often necessary to obtain the covariance matrix of the reconstructed "unprojected" point $\underline{y} \in \mathbb{R}^3$ in Cartesian coordinates. This can be achieved in a closed form using the standard pinhole model [6] and the intrinsic matrix $\mathbf{K}$, which is provided directly by ARKit's API. To give the reader an idea, an example intrinsic matrix from an iPad Pro 2021 has the following form

$$\mathbf{K} = \begin{pmatrix} 212.4 & 0 & 127.0 \\ 0 & 212.4 & 96.3 \\ 0 & 0 & 1 \end{pmatrix}, \qquad (2)$$

which corresponds to a horizontal field of view of approximately 60° and an image size of 256 × 192 pixels. Assuming no uncertainties, the unprojection step can be implemented by introducing a screen space vector in homogeneous coordinates

$$\underline{y}^{uv} = [u, v, 1]^T \qquad (3)$$

which in turn yields

$$\underline{y} = \mathbf{K}^{-1} \cdot \underline{y}^{uv} \cdot z. \qquad (4)$$

We will now extend this step to assume that $\underline{y}^{uv}$ and z are both uncertain. We assume that u and v have a distribution of $\mathcal{U}(-1, 1)$, i.e., they are uniformly distributed between the previous and following pixels. Using moment matching, we obtain

$$\mathbf{C}^{uv} = \operatorname{cov}\left(\underline{y}^{uv}\right) = \operatorname{diag}\left(\frac{1}{3}, \frac{1}{3}, 0\right). \qquad (5)$$

Finally, by using the product rule of independent random variables, we propagate (1) and (5) through (4) to obtain

$$\mathbf{C}^{y} = \mathbf{K}^{-1}\left(\mathbf{C}^{uv}\sigma_z^2 + \underline{y}^{uv}\left(\underline{y}^{uv}\right)^T \sigma_z^2 + \mathbf{C}^{uv} z^2\right)(\mathbf{K}^{-1})^T. \qquad (6)$$

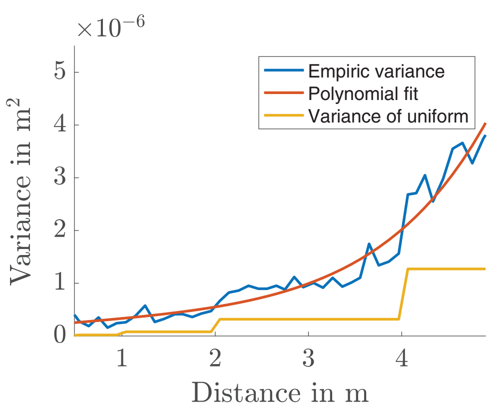
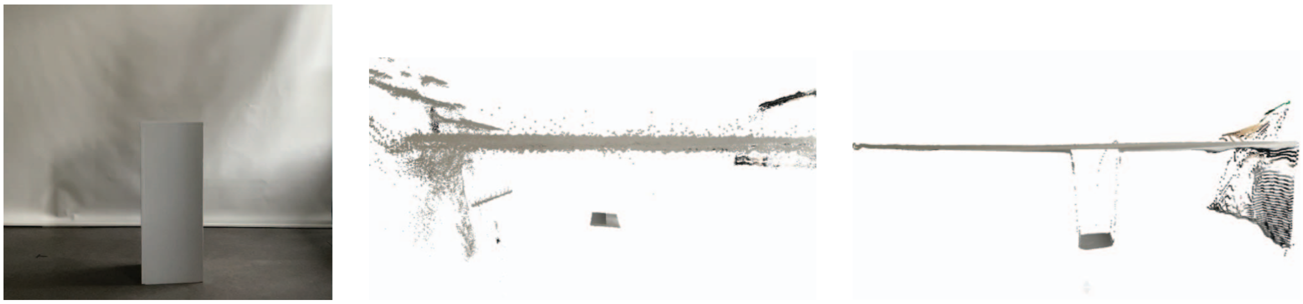


Fig. 10. Empiric variance of measurement noise at different distances, and a best-fitting polynomial fit.

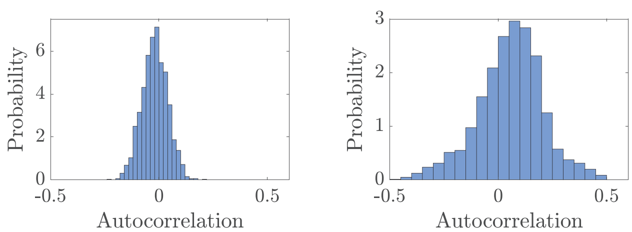
D. Measurement Correlations in Time and Space

When discussing the stochastic properties of measurement noise in sensors, the topic of correlations is usually not mentioned. This omission is justified with sensors such as the Kinect 2, where measurements are mostly independent from each other. However, due to the ML interpolation system, these assumptions cannot be guaranteed to hold for the iPad. Note that, especially in probabilistic estimators, ignoring correlations can lead to estimates with misleading variances, as the estimator cannot compensate for the fact that measurements with dependent noise terms carry less information. This



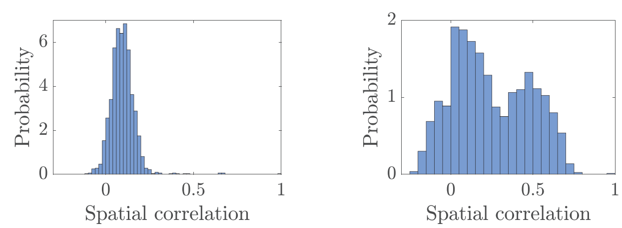
(a) Setup: Cuboid in front of a wall (b) View from above, Kinect 2 (c) View from above, iPad

Fig. 11. Setup of the cuboid in front of wall, observed by a Kinect 2 and an iPad positioned next to each other. Note how the Kinect 2 produces spatially uncorrelated noise, while the iPad produces a smooth, almost flat surface that shifts and deforms each frame. (a) Setup: cuboid in front of a wall. (b) View from above, Kinect 2. (c) View from above, iPad.



(a) Autocorrelations, Kinect 2 (b) Autocorrelations, iPad

Fig. 12. Distribution of autocorrelation coefficients for multiple pixels in a 20×20 px window around the box center and a shift of $\tau = 1$ frame. (a) Autocorrelations, Kinect 2. (b) Autocorrelations, iPad.



(a) Spatial correlations, Kinect 2 (b) Spatial correlations, iPad

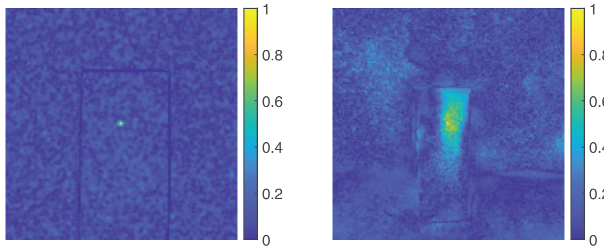
Fig. 13. Distribution of spatial correlation coefficients for multiple pixels in a 20×20 px window in relation to the center pixel. (a) Spatial correlations, Kinect 2. (b) Spatial correlations, iPad.

serves as a motivation to study these correlations more explicitly.

In order to do this, we recorded 150 frames of a rectangular cuboid standing in front of a wall with both an iPad Pro 2021 and a Kinect 2 (see Fig. 11), at a distance of about 2 m, and analyzed a small window of 20×20 px around the center the object. The Kinect 2 will serve as a baseline, as estimators using measurements from this sensor and making no assumptions about dependency have been shown to produce satisfactory reconstructions in [14], [15], and [24]. All of the considered points have a confidence of 2 (high) on the iPad depth image and lie on the cuboid's surface. As an aside, the iPad LIDAR projector was not visible from the Kinect infrared camera, which suggests that both devices do not interfere with each other.

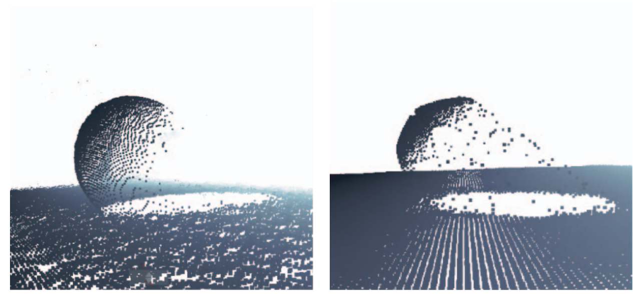
We start with autocorrelations, i.e., how much a series of measurements stemming from the same source is correlated with a time shifted version of itself. This is important given that recursive estimators with time-evolving states, such as the Kalman filter, generally assume that measurements at different time steps are independent. Fig. 12 shows how often an autocorrelation coefficient appears in a 20×20 px window around the box center, for a time shift of $\tau = 1$ frame. For the Kinect 2 [see Fig. 12(a)], we observe that for all positions the autocorrelation coefficient has an absolute value below 0.2, and most of them are below 0.1. However, for the iPad [see Fig. 12(b)], the support of the autocorrelations is wider, briefly reaching 0.5. This shows that iPad measurements (mean 0.034) are much more correlated in time than the Kinect 2 (mean -0.019). This autocorrelation fades with time, taking ten frames for the iPad autocorrelations to reach the same spread as the Kinect. Nonetheless, both are still rather close to 0 most of the time, so we consider it justifiable to assume them as time independent.

Next we will analyze how measurement samples from different pixel positions are spatially correlated. For this, we will take the correlation coefficient of all measurements in the window in relation to the measurements in the center, and tally how often a correlation coefficient appears. The results are shown in Fig. 13. The difference between both sensors, here, is much more remarkable, showing very high correlations across the board for the iPad, with the mean (0.25) being much higher than the Kinect 2 (0.09). This effect can be appreciated visually for the wall in Fig. 11. Here, for the Kinect [see Fig. 11(b), individual measurements can be seen, giving the appearance of a "dusty" cloud, while the iPad depth system generates a smooth surface that moves and deforms between frames [see Fig. 11(c)]. Furthermore, spatial correlations fade much faster for the Kinect as the distance increases. In Fig. 14(a), for example, using a larger 96×96 pixel window, we observe that all measurements farther than 3 px away from the center have a correlation with absolute value below 0.2. For the iPad, however, most of the cuboid surface retains the high correlations.



(a) Spatial correlations, Kinect 2 (b) Spatial correlations, iPad

Fig. 14. Absolute value of the spatial correlations for a 96 × 96 px window, in relation to the pixel in the center. The Kinect 2 image appears "zoomed in" because it has a higher resolution. (a) Spatial correlations, Kinect 2. (b) Spatial correlations, iPad.



(a) Sphere, Kinect 2 (b) Sphere, iPad

Fig. 15. Plastic sphere of radius 15 cm lying on the ground, being observed from a distance of 2 m by a Kinect 2 and an iPad depth sensor. (a) Sphere, Kinect 2. (b) Sphere, iPad.

If desired, spatial correlations can be incorporated into linear estimators, such as the Kalman filter and its extensions, quite easily by introducing a composite measurement and adjusting the measurement equation. Thus, if two measurements $\underline{y}_1$ and $\underline{y}_2$ are "close" to each other and known to stem from the same surface, the estimator can use

$$\underline{y} = \begin{bmatrix} \underline{y}_1 \\ \underline{y}_2 \end{bmatrix} \tag{7}$$

instead, with covariance matrix

$$\mathbf{C}^y = \begin{pmatrix} \mathbf{C}_1^y & \mathbf{C}_{1,2}^y \\ \left(\mathbf{C}_{1,2}^y\right)^T & \mathbf{C}_2^y \end{pmatrix}. \tag{8}$$

Here, $\mathbf{C}_1^y$ and $\mathbf{C}_2^y$ are calculated as before from (6). In order to derive $\mathbf{C}_{1,2}^y$, we assume that the y^{uv} components are independent from each other and from z_1 and z_2. Furthermore, it holds that

$$\operatorname{cov}(z_1, z_2) = \sigma_{z,1}\sigma_{z,2} \operatorname{corr}(z_1, z_2) \tag{9}$$

where $\operatorname{corr}(z_1, z_2)$ is the scene-dependent spatial correlation coefficient. Our experiments have shown that $\operatorname{corr}(z_1, z_2)$ usually hovers around 0.2 if the distance is less than 20 px and both measurements stem from the same surface. Finally, the correlation matrix $\mathbf{C}_{1,2}^y$ can be obtained in closed form yielding

$$\mathbf{C}_{1,2}^y = \operatorname{cov}(\underline{y}_1, \underline{y}_2) \tag{10}$$

$$= \operatorname{cov}\left(\mathbf{K}^{-1} \cdot \underline{y}_1^{uv} \cdot z_1, \mathbf{K}^{-1} \cdot \underline{y}_2^{uv} \cdot z_2\right) \tag{11}$$

$$= \left(\mathbf{K}^{-1} \cdot \underline{y}_1^{uv} \left(\underline{y}_2^{uv}\right)^T \left(\mathbf{K}^{-1}\right)^T\right) \operatorname{cov}(z_1, z_2). \tag{12}$$

E. Working with Curved Surfaces

In the previous sections, we showed that the iPad produced relatively accurate measurements from large flat surfaces, such as walls. However, due to the low spatial resolution, smaller objects (in relation to the field of view) will appear ''smoothened" out, with hard corners flattened and surfaces fused together. This effect can be compensated by ensuring that the object size is much larger than 11 px, as explained in Section II-B. Still, the tendency to flatten or merge surfaces can become problematic in applications that require as much knowledge as possible about the target's shape, such as extended object tracking and classification.

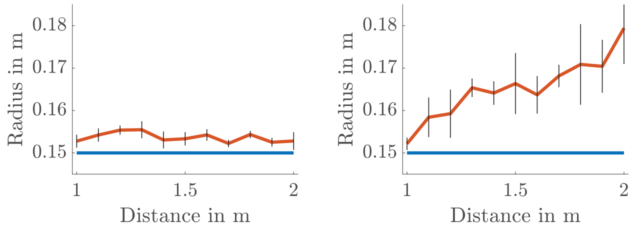
In this section, we will analyze how much information about the shape is lost at different distances by estimating the extent and position of sphere. A description of the experimental setup follows. The target sphere, with a ground-truth radius of 15 cm, was already introduced in the scene from Fig. 2. Here, instead, we place it on the floor, as shown in the example captures from Fig. 15, and observe it from a height of approximately 1.5 m. As a note, in these images, we can also appreciate the contrast between the noise correlations mentioned in Section III-D: with the Kinect 2, the floor has a highly irregular texture, whereas with the iPad, it appears almost perfectly flat.

The estimation procedure is as follows. For a given frame, the sphere is represented using the following state:

$$\underline{x} = [\underline{p}^T, r]^T \in \mathbb{R}^4, \tag{13}$$

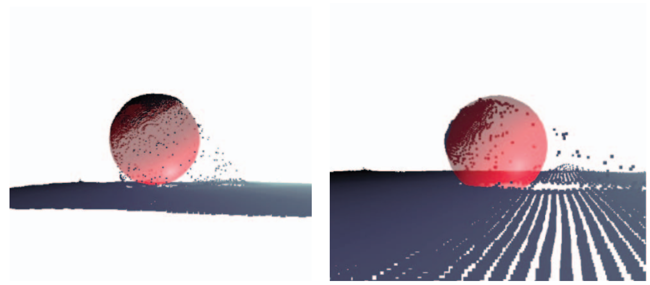
where $\underline{p}$ is the position in $\mathbb{R}^3$ and r is the radius. The following preprocessing steps are executed. First, the screen measurements are unprojected into $\mathbb{R}^3$ using (4) and (6). Second, the ground plane is estimated using RANSAC, and all pixels that belong to it with a threshold of 2 cm are removed. Third, we also eliminate fringe measurements (flying pixels), defined here as any pixel with at least one neighbor farther than 2 cm away. Finally, we use the remaining n measurements $\underline{y}_i$ for $1 \leq i \leq n$ to estimate the state $\underline{x}$ using least squares shape fitting. Here, the idea is to minimize the weighted sum of the squared residuals

$$R_i^2 = \left\|\underline{y}_i - \underline{p}\right\|^2 - r^2, \tag{14}$$



(a) Estimated radius, Kinect 2 (b) Estimated radius, iPad

Fig. 16. Estimated radius of the sphere for the Kinect 2 and the iPad, viewed from different camera distances. Ground truth in blue, result means in red, and standard deviations in black (vertical lines). (a) Estimated radius, Kinect 2. (b) Estimated radius, iPad.



(a) Sphere at 1 m (b) Sphere at 2 m

Fig. 17. Point cloud of the plastic sphere being observed by the iPad depth sensor. Best-fitting sphere overlaid in red. Floor appears wider at 2 m due to the field of view. (a) Sphere at 1 m. (b) Sphere at 2 m.

where $|| \cdot ||^2$ is the square of the Euclidean norm, using the inverse of the residual variances as weights

$$w_i = \frac{1}{\mathrm{var}(R_i^2)} = \frac{1}{\mathrm{tr}(\mathbf{C}_i^y)}. \tag{15}$$

The resulting $\underline{x}$ can be obtained with standard nonlinear minimization. This state estimation procedure is repeated along multiple frames as the camera moves horizontally from a distance of 1 to 2 m away from the sphere. For the sake of simplicity, the measurements are assumed to be independent from each other.

Fig. 16 shows the results at distances in intervals of 10 cm. The ground truth of 15 cm in blue, the means are shown in dark red, and the standard deviations in vertical red lines. Note that, due to the presence of artifacts, the size of the sphere can vary moderately even in consecutive frames. Furthermore, by eliminating fringe pixels, we remove measurements around the sphere border. Thus, it would be expected for the estimated radius to be lower than the ground truth, which stands in contrast with both results. This effect, however, is compensated in the opposite direction by the extent bias caused by measurement noise, a known effect in shape fitting studied in [9] and [25] among others. This bias appears constant for the Kinect 2 [see Fig. 16(a)], where the radius is consistently around 5 mm (3%) higher than the ground truth. However, the iPad, while producing accurate results at around 1 m, quickly starts losing accuracy [see Fig. 16(b)] as the camera moves away. This is a consequence of the sensor flattening the measured surface as it merges into the floor, an effect also observed in previous experiments, which in turn increases the size of the estimated sphere.

Example results can be seen in Fig. 17, with the estimated sphere in red. Here, we can see how, at a short distance, measurement quality is comparable to the Kinect in Fig. 15(a), yielding a radius estimate of 15.5 cm. However, after a short distance, measurements become more sparse and the proportion of fringe pixels increases significantly. Furthermore, the patch becomes so flattened that the increased radius (17.5 cm) pushes the sphere position into the ground. Note that, at this distance, the sphere is only 32 px wide. After this point, the radius reaches 20 cm at a distance of 2.3 m, and beyond that segmentation starts to become difficult, given much of the sphere has merged into the floor.

IV. CONCLUSION

In this paper, we presented a quantitative analysis of the spatial uncertainties for the iPad Pro depth sensor. As motivation, we explained how Apple, with its high market share in mobile devices, has begun shipping depth sensors integrated in their smartphones and tablets, increasing the reach of applications in localization, tracking, and classification without extra cost to developers and users. Thus, it makes sense to analyze the properties, benefits, and pitfalls of these new data streams. First, we briefly described the direct ToF sensor that the tablet uses to obtain depth images, and pointed out that the depth stream is most likely extrapolated from only 576 real measurements, which would explain the observed low spatial resolution. We also noted that the values provided by the API are not the direct measured values, and instead, they are generated by an ML algorithm that incorporates information from the color stream. As part of the analysis, we measured the discretization of the depth domain, provided a model for the measurement bias and error variance, and described the temporal and spatial correlations between measurements. We also showed the tendency of the iPad depth sensor to merge and flatten surfaces, which is useful when dealing with planes such as walls, but becomes problematic when estimating the shape of curved objects, such as spheres.

In general, it can be seen that the iPad depth sensor, in its current iteration, is well suited for simultaneous localization and mapping (SLAM) based on planar surfaces. This is shown by the robust camera tracking provided by the default libraries. Dealing with other objects, however, imposes some restrictions on their size and how far they can be from the sensor. Curved objects, or objects with steep depth changes, appear flattened out, which can reduce its applicability in fields, such as extended object tracking or object classification based on point clouds, which have higher requirements on measurement quality. Thus, applications that deal with these objects should keep in mind the tightened operating range. Still, these limitations should be balanced with the advantages provided by the Apple ecosystem, and the wide reach of potential users available to applications using it.

REFERENCES

[1] 4da.tech, "LiDAR: Apple LiDAR Analysis," 2021. Accessed: Oct. 26, 2021. [Online]. Available: https://3da.medium.com/lidar-apple-lidar-and-dtof-analysis-cc18056ec41a.

[2] Apple, "iPad Pro 2020," Accessed: Oct. 26, 2021. [Online]. Available: https://www.apple.com/de/ipad-pro/

[3] Apple, "ARKit," 2021. Accessed: Oct. 26, 2021. [Online]. Available: https://developer.apple.com/augmented-reality/

[4] A. Breitbarth, T. Schardt, C. Kind, J. Brinkmann, P.-G. Dittrich, and G. Notni "Measurement accuracy and dependence on external influences of the iPhone X TrueDepth sensor," in Proc. Photonics Educ. Meas. Sci., 2019, pp. 1114407.

[5] A. Corti, S. Giancola, G. Mainetti, and R. Sala "A metrological characterization of the Kinect v2 time-of-flight camera," *Robot. Auton. Syst.*, vol. 75, pp. 584–594, 2016.

[6] K. M. Dawson-Howe and D. Vernon "Simple pinhole camera calibration," *Int. J. Imag. Syst. Technol.*, vol. 5, no. 1, pp. 1–6, 1994.

[7] D. Eigen, C. Puhrsch, and R. Fergus "*Depth map prediction from a single image using a multi-scale deep network*," in Proc. Proc. 27th Int. Conf. Neural Inf. Process. Syst., 2014, pp. 2366–2374.

[8] F. Faion, S. Friedberger, A. Zea, and U. D. Hanebeck "Intelligent sensor-scheduling for multi-Kinect-tracking," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2012, pp. 3993–3999.

[9] F. Faion, A. Zea, and U. D. Hanebeck "Reducing bias in Bayesian shape estimation," in *Proc. 17th Int. Conf. Inf. Fusion (FUSION)*, 2014, pp. 1–8.

[10] R. Garg, V. K. Bg, G. Carneiro, and I. Reid "Unsupervised CNN for single view depth estimation: Geometry to the rescue," in Proc. Eur. Conf. Comput. Vis., 2016, pp. 740–756.

[11] C. Godard, O. Mac Aodha, and G. J. Brostow "Unsupervised monocular depth estimation with left-right consistency," in *Proc. IEEE Conf. Comput. Vis. pattern Recognit.*, 2017, pp. 270–279.

[12] C. Gollob, T. Ritter, R. Krabnitzer, A. Tockner, and A. Nothdurft "Measurement of forest inventory parameters with Apple iPad pro and integrated LiDAR technology," *Remote Sens.*, vol. 13, no. 16, pp. 3129, 2021.

[13] Google, "ARCore," 2021. Accessed: Oct. 26, 2021. [Online]. Available: https://developers.google.com/ar

[14] S. Izadi et al., "Kinectfusion: Real-time 3D reconstruction and interaction using a moving depth camera," in *Proc. 24th Annu. ACM Symp. User Interface Softw. Technol.*, 2011, pp. 559–568. [Online].

[15] S.-Y. Jiang, N. Y.-C. Chang, C.-C. Wu, C.-H. Wu, and K.-T. Song "Error analysis and experiments of 3D reconstruction using a RGB-D sensor," in *Proc. IEEE Int. Conf. Automat. Sci. Eng.*, 2014, pp. 1020–1025.

[16] N. Kitsunezaki, E. Adachi, T. Masuda, and J. Mizusawa "Kinect applications for the physical rehabilitation," in *Proc. IEEE Int. Symp. Med. Meas. Appl.*, 2013, pp. 294–299.

[17] G. Luetzenburg, A. Kroon, and A. A. Bjork "Evaluation of the Apple iPhone 12 pro LiDAR for an application in geosciences," *Sci. Rep.*, vol. 11, no. 1, pp. 1–9, 2021.

[18] L. Gregor, K. Aart, and B. Anders, "Lidar scanning of coastal cliffs in Denmark and Greenland with the Apple iPhone 12 pro," in *Proc. AGU Fall Meeting 2021*, 2021.

[19] R. Lun and W. Zhao "A survey of applications and human motion recognition with Microsoft Kinect," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 29, no. 05, 2015, Art. no. 1555008.

[20] E. Marder-Eppstein "Project tango," in *Proc. ACM SIGGRAPH 2016 Real-Time Live!*, 2016, Art. no. 40.

[21] A. Matsuura, M. Dan, A. Hirano, Y. Kiku, S. Torii, and S. Morita "Body measurement of riding horses with a versatile tablet-type 3D scanning device," *J. Equine Sci.*, vol. 32, no. 3, pp. 73–80, 2021.

[22] I. Moazzam, K. Kamal, S. Mathavan, S. Usman, and M. Rahman "Metrology and visualization of potholes using the Microsoft Kinect sensor," in *Proc. 16th Int. IEEE Conf. Intell. Transp. Syst.*, 2013, pp. 1284–1291.

[23] A. Murtiyoso, P. Grussenmeyer, T. Landes, and H. Macher "First assessments into the use of commercial-grade solid state LIDAR for low cost heritage documentation," *ISPRS-Int. Archives Photogrammetry Remote Sens. Spatial Inf. Sci.*, vol. 43, pp. 599–604, 2021.

[24] C. V. Nguyen, S. Izadi, and D. Lovell "Modeling Kinect sensor noise for improved 3D reconstruction and tracking," in *Proc. Second Int. Conf. 3D Imaging, Model., Process., Visual. Transmiss.*, 2012, pp. 524–530.

[25] T. Okatani and K. Deguchi "On bias correction for geometric parameter estimation in computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 959–966.

[26] J.-H. Park, Y.-D. Shin, J.-H. Bae, and M.-H. Baeg "Spatial uncertainty model for visual features using a Kinect sensor," *Sensors*, vol. 12, no. 7, pp. 8640–8662, 2012.

[27] A. Spreafico, F. Chiabrando, L. Teppati Lose, and F. Giulio Tonolo "The iPad pro built-in LIDAR sensor: 3D rapid mapping tests and quality assessment," *ISPRS-Int. Archives Photogrammetry Remote Sens. Spatial Inf. Sci.*, vol. 43, pp. 63–69, 2021.

[28] *System Plus Consulting*, "Apple iPad Pro LiDAR module," 2021. Accessed: Oct. 26, 2021. [Online]. Available: https://www.systemplus.fr/reverse-costing-reports/apple-ipad-pro-11s-lidar-module/

[29] S. Tatsumi, K. Yamaguchi, and N. Furuya "ForestScanner: A mobile application for measuring and mapping trees with LiDAR-equipped iPhone and iPad," *Methods in Ecology and Evolution*, 2021. https://besjournals.onlinelibrary.wiley.com/doi/abs/10.1111/2041-210X.13900

[30] M. Vogt, A. Rips, and C. Emmelmann "Comparison of iPad Pros LiDAR and TrueDepth capabilities with an industrial 3D scanning solution," Technologies, vol. 9, no. 2, 2021. Art. no. 25. [Online]. Available: https://www.mdpi.com/2227-7080/9/2/25

[31] X. Wang
et al., "Evaluation of ipad pro 2020 LIDAR for estimating tree diameters in urban forest," ,
ISPRS Ann. Photogrammetry Remote Sens. Spatial Inf. Sci., vol. 8, pp. 105–110, 2021.

[32] O. Wasenmuller and D. Stricker
"Comparison of Kinect v1 and v2 depth images in terms of accuracy and precision,"
in *Proc. Asian Conf. Comput. Vis.*, 2016, pp. 34–45.

[33] X. Xu, A. Al-Dahle, and K. Garg
"Shared sensor data across sensor processing pipelines,"
U. S. Patent 10 671 068, 6 2, 2020.

Dr.-Ing. Antonio Zea is a researcher at the Intelligent Sensor-Actuator-Systems (ISAS) Laboratory of the Karlsruhe Institute of Technology (KIT), where he also obtained his Ph.D. degree in the field of Computer Science in 2018. His research interests include extended object tracking, localization, robot teleoperation with virtual and augmented Reality, and human machine interaction.

Uwe D. Hanebeck is currently a Chaired Professor of computer science with the Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany, and the Director of the Intelligent Sensor-Actuator-Systems Laboratory (ISAS), Karlsruhe, Germany. His research interests include information fusion, nonlinear state estimation, stochastic modeling, system identification, and control with a strong emphasis on theory-driven approaches based on stochastic system theory and uncertainty models.

Split Happens! Imprecise and Negative Information in Gaussian Mixture Random Finite Set Filtering

KEITH A. LEGRAND
SILVIA FERRARI

In object-tracking and state-estimation problems, ambiguous evidence such as imprecise measurements and the absence of detections can contain valuable information and thus be leveraged to further refine the probabilistic belief state. In particular, knowledge of a sensor's bounded field of view (FoV) can be exploited to incorporate evidence of where an object was not observed. This paper presents a systematic approach for incorporating knowledge of the FoV geometry, position, and object inclusion/exclusion evidence into object state densities and random finite set multiobject cardinality distributions. The resulting state estimation problem is nonlinear and solved using a new Gaussian mixture approximation based on recursive component splitting. Based on this approximation, a novel Gaussian mixture Bernoulli filter for imprecise measurements is derived and demonstrated in a tracking problem using only natural language statements as inputs. This paper also considers the relationship between bounded FoVs and cardinality distributions for a representative selection of multiobject distributions, which can be used for sensor planning, as is demonstrated through a problem involving a multi-Bernoulli process with up to 100 potential objects.

Manuscript received July 21, 2022; revised October 13, 2022; released for publication March 20, 2023.

Keith A. LeGrand is with the Sensing, Controls, and Probabilistic Estimation (SCOPE) Laboratory, School of Aeronautics and Astronautics, Purdue University, West Lafayette, IN 47907, USA. Silvia Ferrari is with the Laboratory for Intelligent Systems and Controls (LISC), Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca 14853.

Refereeing of this contribution was handled by Florian Meyer.

This work was supported in part by the Office of Naval Research Grant N0014-19-1-2266 and by the National Defense Science and Engineering Graduate (NDSEG) Fellowship.

1557-6418/22/$17.00 © 2022 JAIF

I. INTRODUCTION

Random finite set (RFS) theory has been proven a highly effective framework for developing and analyzing tracking and sensor planning algorithms in applications involving an unknown number of multiple targets (objects) [1]–[7]. Until recently, however, little attention has been devoted to the role that bounded sensor fields of view (FoVs) play in assimilating measurements, or lack thereof, into multiobject probability distributions. Existing tracking algorithms, for example, typically terminate object tracks when the object leaves the sensor field-of-view (FoV). While this approach is suitable when the FoV doubles as the tracking region of interest (ROI), it is inapplicable when the sensor FoV is much smaller than the ROI and, thus, must be moved or positioned so as to maximize information value [8]–[13]. Other technical challenges arise in multisensor fusion problems involving bounded overlapping FoVs and have been the focus of recent work[14]–[17].

The simple indication of an object's presence or absence within a known region, such as an FoV, is powerful evidence that can be incorporated to update the object probability density function (pdf) in a Bayesian framework. For example, the absence of detections is a type of *negative information* indicating that the object state may reside outside the FoV [18]. In contrast, binary-type sensors may produce *imprecise measurements* [19]–[21] that indicate the object is inside the sensor FoV but provide no further localization information. Similarly, "soft" data from human sources, such as natural language statements, can be considered as imprecise measurements due to their inherent ambiguity [22], [23]. Particle-based filtering algorithms [21], [24], [25] can accommodate such measurements but require a large number of particles and are computationally expensive. The integrated track splitting filter for state-dependent probability of detection (ITSpd) [26] uses Gaussian mixtures (GMs) to model both the object pdf and the state-dependent probability of detection function. Though GMs efficiently model some detection probability functions, other simple functions, such as uniform probability densities over a 3D FoV, require problematically large numbers of components. Other approaches [27], [28] employ stochastic sampling and the expectation maximization (EM) algorithm to compute GM approximations to the posterior pdf. However, the use of intermediate particle representations and EM reconstruction can lead to information loss, and convergence is sensitive to EM initial condition specification.

This paper presents new methods for incorporating inclusion/exclusion evidence in Bayesian single-object and multiobject estimation and sensor planning algorithms, as illustrated in Fig. 1. Section II defines the notation used in this paper, and Section III details the problem formulation and related assumptions. Section IV presents a deterministic method that partitions a GM state density along the boundaries of a known

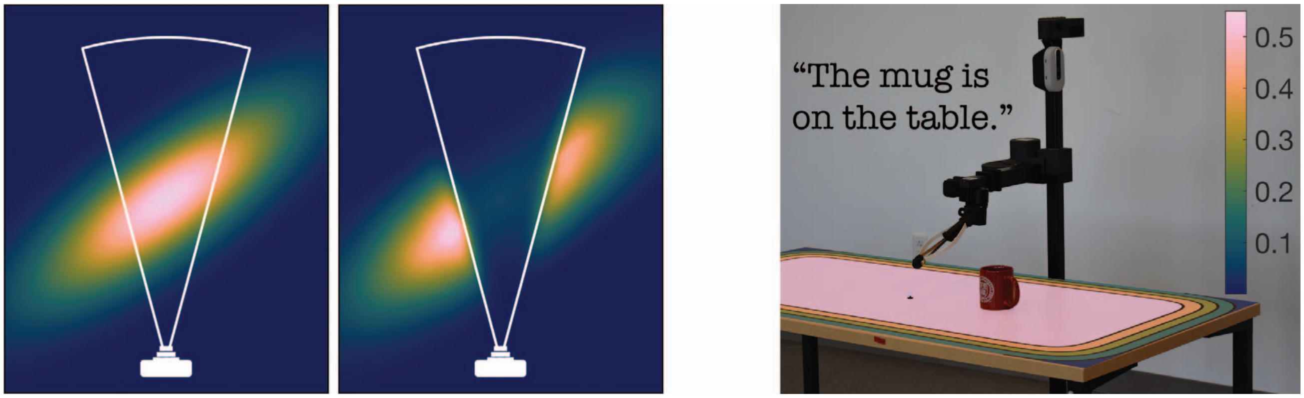


Fig. 1. Gaussian mixture probability distribution before (left) and after (center) incorporating negative information (that is, the absence of detections) and the known bounded sensor FoV, and a Gaussian mixture distribution after incorporating an imprecise measurement corresponding to a set of possible mug locations (right) in a robot perception application.

region, such as an FoV, through recursive Gaussian splitting. By this approach, negative information is leveraged in GM filters to further refine the posterior object state pdf. Similarly, imprecise measurements, such as natural language statements, can be incorporated to obtain GM posterior distributions using a new multi-FoV-generalized splitting algorithm. Section V presents an application of the splitting method to the tracking of a person in a crowded space using natural language statements and a new GM Bernoulli filter algorithm. In Section VI, FoV object cardinality probability mass functions (pmfs) are derived for some of the most commonly encountered RFS distributions. Section VII presents an application of bounded FoV statistics to a sensor placement problem, and conclusions are made in Section VIII. This paper builds on previous work [29] by presenting a generalized partitioning algorithm for use with multiple FoVs, a derivation of a new GM Bernoulli filter algorithm applicable to imprecise measurements, and a simulation of a tracking problem using natural language statements.

II. NOTATION

Throughout this paper, single-object states are represented by lowercase letters (e.g., $\mathbf{x}, \mathring{x}$), while multiobject states are represented by italic uppercase letters (e.g., X, $\mathring{X}$). Bold lowercase letters are used to denote vectors, and bold uppercase letters are used denote matrices. The accent "$\circ$" is used to distinguish labeled states and functions (e.g., $\mathring{f}, \mathring{x}, \mathring{X}$) from their unlabeled equivalents (e.g., f, x, X). Spaces are represented by blackboard-bold symbols (e.g., $\mathbb{X}, \mathbb{L}$).

The multiobject exponential notation,

$$h^A \triangleq \prod_{a \in A} h(a), \tag{1}$$

where $h^{\emptyset} \triangleq 1$ is adopted throughout. For multivariate functions, the dot $(\cdot)$ denotes the argument of the multiobject exponential, e.g.,:

$$[g(a, \cdot, c)]^B \triangleq \prod_{b \in B} g(a, b, c). \tag{2}$$

The exponential notation is used to denote the product space, $\mathbb{X}^n = \prod^n (\mathbb{X} \times)$, whereas exponents of RFSs are used to denote RFSs of a given cardinality, e.g., $|X^n| = n$, where n is the cardinality. The set of natural numbers less than or equal to n is denoted by

$$\mathbb{N}_n \triangleq \{1, \ldots, n\}. \tag{3}$$

The operator $\mathrm{diag}(\cdot)$ places its input on the diagonal of the zero matrix. The Kronecker delta function is defined as

$$\delta_{\boldsymbol{a}}(\boldsymbol{b}) \triangleq \begin{cases} 1, \text{ if } \boldsymbol{b} = \boldsymbol{a} \\ 0, \text{ otherwise} \end{cases} \tag{4}$$

for any two arbitrary vectors $\boldsymbol{a}$, $\boldsymbol{b} \in \mathbb{R}^n$. The inner product of two integrable functions $f(\cdot)$ and $g(\cdot)$ is denoted by

$$\langle f, g \rangle = \int f(\mathbf{x}) g(\mathbf{x}) \mathrm{d}\mathbf{x}. \tag{5}$$

III. PROBLEM FORMULATION AND ASSUMPTIONS

This paper considers the incorporation of inclusion/exclusion evidence into algorithms for (multi)object tracking and sensor planning when the number of objects is unknown and time-varying. Often in tracking, object detection may depend only on a partial state $\mathbf{s} \in \mathbb{X}_s \subseteq \mathbb{R}^{n_s}$, where $\mathbb{X}_s \times \mathbb{X}_v = \mathbb{X} \subseteq \mathbb{R}^{n_x}$ forms the full object state space. For example, the instantaneous ability of a sensor to detect an object may depend only on the object's relative position. In that case, $\mathbb{X}_s$ is the position space, and $\mathbb{X}_v$ is comprised of nonposition states, such as object velocity. This nomenclature is adopted throughout the paper, while noting that the approach is applicable to other state definitions. Following [30], the sensor FoV can be defined as the compact subset $\mathcal{S}(\boldsymbol{q}) \subset \mathbb{X}_s$. In general, the FoV is a function of the sensor state $\boldsymbol{q}$, which, for example, may consist of the sensor position, orientation, and zoom level. However, for notational simplicity, this dependence is omitted in the remainder of this paper.

Now, let the object state $\mathbf{x}$ consist of the kinematic variables that are to be estimated from data via filtering, such as the object position, velocity, and turn rate. Then, the single-object pdf is denoted by $p(\mathbf{x})$. Letting $\mathbf{s} = \text{proj}_{\mathbb{X}_s}\mathbf{x}$ denote the state elements that correspond to $\mathbb{X}_s$, an object's presence inside the FoV can be expressed by the generalized indicator function

$$1_{\mathcal{S}}(\mathbf{x}) = \begin{cases} 1, & \text{if } \mathbf{s} \in \mathcal{S} \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

The number of objects and their kinematic states are unknown *a priori*, but can be assumed to consist of discrete and continuous variables, respectively. The collection of object states is modeled as an RFS X or labeled random finite set (LRFS) $\mathring{X}$, where the single-object labeled state $\mathring{x} = (\mathbf{x}, \ell) \in \mathbb{X} \times \mathbb{L}$ consists of a kinematic state vector $\mathbf{x}$ and a unique discrete label ℓ. It is assumed that the prior multiobject distribution is known, e.g., from the output of a multiobject filter, and modeled using either the RFS density $f(X)$ or the LRFS density $\mathring{f}(\mathring{X})$.

In RFS-based tracking, single-object densities are, in fact, parameters of the higher-dimensional multiobject density. Non-Gaussian single-object state densities are often modeled using GMs because they admit closed-form approximations to the multiobject Bayes recursion under certain conditions [2], [31]. Therefore, in this paper, it is assumed that single-object densities (which are parameters of the higher-dimensional multiobject density) are parameterized as

$$p(\mathbf{x}) = \sum_{\ell=1}^{L} w^{(\ell)} \mathcal{N}(\mathbf{x};\ \mathbf{m}^{(\ell)},\ \mathbf{P}^{(\ell)}), \tag{7}$$

where L is the number of GM components and $w^{(\ell)}, \mathbf{m}^{(\ell)}$, and $\mathbf{P}^{(\ell)}$ are the weight, mean, and covariance matrix of the ℓth component, respectively.

In this paper, the problem considered is forming GM Bayesian posteriors given evidence of the forms:

- T1 *The existence or nonexistence of a measurement is evidence of the inclusion or exclusion of the object state within a known set.* For example, the nonexistence of a detection (measurement) is evidence of an object's position exclusion from the sensor FoV.
- T2 *The value of the measurement is evidence of the inclusion or exclusion of the object state within a known set*. For example, the observation that a sea-level freshwater lake is frozen is evidence that the water temperature belongs to the set of temperatures below 0 °C.

Mahler's finite-set statistics (FISST) provides the mathematical foundation for modeling types T1 and T2 using state-dependent probability of detection functions and generalized likelihood functions, respectively. However, in both cases, the Bayes posterior involves products of the prior GM with indicator functions such as

$$p(\mathbf{x})1_{\mathcal{S}}(\mathbf{x}) \triangleq p_{\mathcal{S}}(\mathbf{x}) \qquad \text{and} \tag{8}$$

$$(1 - 1_{\mathcal{S}}(\mathbf{x}))p(\mathbf{x}) \triangleq p_{\mathcal{C}(\mathcal{S})}(\mathbf{x}), \tag{9}$$

where $\mathcal{C}(\mathcal{S})$ denotes the complement space $\mathbb{X}_s \backslash \mathcal{S}$. Thus, the resulting posterior is no longer a GM.

This paper presents a fast GM approximation of (8) and (9), thereby enabling the assimilation of inclusion/exclusion evidence in any GM-based RFS single-object or multiobject filter. Building on these concepts, this paper also considers the role of inclusion/exclusion evidence in object cardinality distributions and derives pmf expressions that describe the probabilities associated with different numbers of objects existing within a given set $\mathcal{S}$ (such as an FoV).

IV. GM APPROXIMATION OF FOV-PARTITIONED DENSITIES

This section presents a method for partitioning the object pdf into truncated densities $p_{\mathcal{S}}(\mathbf{x})$ and $p_{\mathcal{C}(\mathcal{S})}(\mathbf{x})$ with supports $\mathcal{S} \times \mathbb{X}_v$ and $\mathcal{C}(\mathcal{S}) \times \mathbb{X}_v$, respectively. Focus is given to the single-object state density with the awareness that the method is naturally extended to RFS multiobject densities and algorithms that use GM parameterization.

Consider the single-object density $p(\mathbf{x})$ parameterized by an L-component GM, as follows:

$$p(\mathbf{x}) = p_{\mathcal{S}}(\mathbf{x}) + p_{\mathcal{C}(\mathcal{S})}(\mathbf{x}) = \sum_{\ell=1}^{L} w^{(\ell)} \mathcal{N}(\mathbf{x};\ \mathbf{m}^{(\ell)}, \mathbf{P}^{(\ell)}). \tag{10}$$

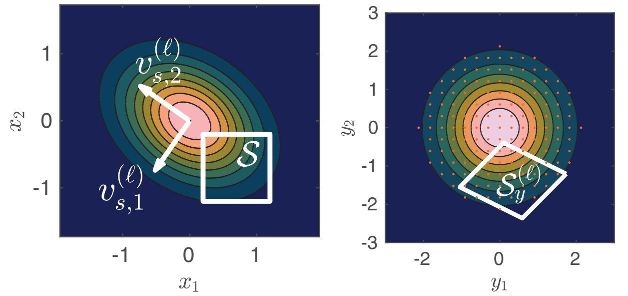
One simple approximation of densities partitioned according to the discrete FoV geometry, referred to as FoV-partitioned densities hereon, is found by evaluating the indicator function at the component means [32], i.e.,:

$$p_{\mathcal{S}}(\mathbf{x}) \approx \sum_{\ell=1}^{L} w^{(\ell)} 1_{\mathcal{S}}(\mathbf{m}^{(\ell)}) \mathcal{N}(\mathbf{x};\ \mathbf{m}^{(\ell)},\ \mathbf{P}^{(\ell)}), \tag{11}$$

$$p_{\mathcal{C}(\mathcal{S})}(\mathbf{x}) \approx \sum_{\ell=1}^{L} w^{(\ell)} (1 - 1_{\mathcal{S}}(\mathbf{m}^{(\ell)})) \mathcal{N}(\mathbf{x};\ \mathbf{m}^{(\ell)},\ \mathbf{P}^{(\ell)}). \tag{12}$$

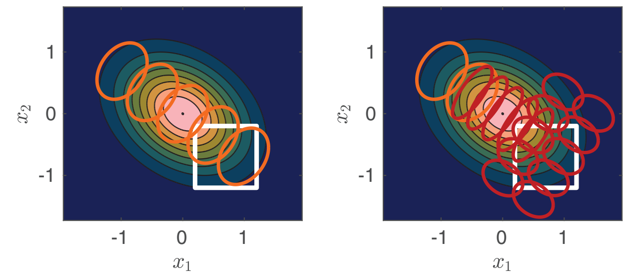
By this approach, components whose means lie inside (outside) the FoV are preserved (pruned), or vice versa.

The accuracy of this mean-based partition approximation depends strongly on the resolution of the GM near the geometric boundaries of the FoV. Even though the mean of a given component lies inside (outside) the FoV, a considerable proportion of the probability mass may lie outside (inside) the FoV, as is illustrated in Fig. 2(a). Therefore, the amount of FoV overlap, along with the weight of the component, determines the accuracy of the approximations (11) and (12). To that end,



(a) (b)

Fig. 2. Original component density and FoV with covariance eigenvectors overlaid (a), and same component density and FoV after change of variables (b).



(a) (b)

Fig. 3. 1σ contours of components after first-split operation (a) and second-split operation (b), where components formed in the second operation are shown in red.

the algorithm presented in the following subsection iteratively resolves the GM near FoV bounds by recursively splitting Gaussian components that overlap the FoV bounds.

A. Gaussian Splitting Algorithm

The Gaussian splitting algorithm presented in this subsection forms an FoV-partitioned GM approximation of the original GM by using a higher number of components near the FoV boundaries, $\partial\mathcal{S}$, so as to improve the accuracy of the mean-based partition.

Consider for simplicity a two-dimensional example in which the original GM, $p(\mathbf{x})$, has a single component whose mean lies outside the FoV, as shown in Fig. 2(a). The algorithm first applies a change of variables $\mathbf{x} \mapsto \mathbf{y} \in \mathbb{Y} \subseteq \mathbb{R}^{n_s}$ such that $p(\mathbf{y})$ is symmetric and has a zero mean and unit variance. The basis vectors of the space $\mathbb{Y}$ correspond to the principal directions of the component's position covariance. The same change of variables is applied to the FoV bounds [Fig. 2(b)].

A pre-computed point grid is then tested for inclusion in the transformed FoV in order to decide whether to split the component and, if so, along which principal direction. For each new split component, the process is repeated—if a new component significantly overlaps the FoV boundaries, it may be further split into several smaller components, as illustrated in Fig. 3. This process is repeated until the stopping criteria are satisfied. After the GM splitting terminates, $p_{\mathcal{S}}(\mathbf{x})$ and $p_{\mathcal{C}(\mathcal{S})}(\mathbf{x})$ are approximated by the mean-based partition [(8) and (9)], as illustrated in Fig. 4.

B. Univariate Splitting Library

Splitting is performed efficiently by utilizing a pregenerated library of optimal split parameters for the univariate standard Gaussian $q(x)$, as first proposed in [33] and later generalized in [34]. The univariate split parameters are retrieved at run-time and applied to arbitrary multivariate Gaussian densities via scaling, shifting, and covariance diagonalization.

Generation of the univariate split library is performed by minimizing the cost function

$$J = L_2(q||\tilde{q}) + \lambda\tilde{\sigma}^2 \qquad \text{s.t.} \sum_{j=1}^{R} \tilde{w}^{(j)} = 1, \tag{13}$$

where

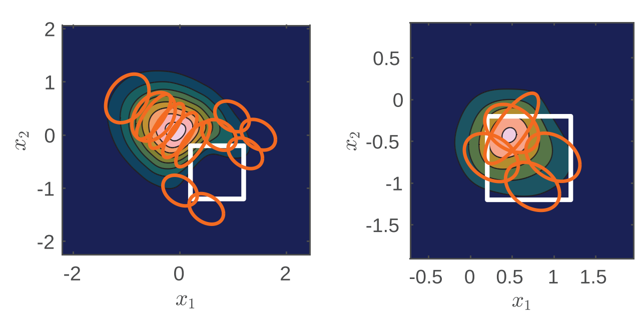
$$\tilde{q}(x) = \sum_{j=1}^{R} \tilde{w}^{(j)} \mathcal{N}(x;\, \tilde{m}^{(j)},\, \tilde{\sigma}^2) \tag{14}$$

for different parameter values R, λ. The regularization term λ balances the importance of using smaller standard deviations $\tilde{\sigma}$ with the minimization of the L_2 distance. While other distance measures may be used, the L_2 distance is attractive because it can be computed in closed form for GMs [34]. As an example, the optimal split parameters for $R = 4, \lambda = 0.001$ are provided in Table I.

C. Change of Variables

The determination of which components should be split and, if so, along which direction, is simplified by first establishing a change of variables. For each component with index ℓ, the change of variables $\boldsymbol{h}^{(\ell)} : \mathbb{X}_s \mapsto \mathbb{Y}$ is applied as follows:

$$\mathbf{y} = \boldsymbol{h}^{(\ell)}(\mathbf{s}; \mathbf{m}_s^{(\ell)}, \mathbf{P}_s^{(\ell)}) \triangleq (\boldsymbol{\Lambda}_s^{(\ell)})^{-\frac{1}{2}} \boldsymbol{V}_s^{(\ell)T}(\mathbf{s} - \mathbf{m}_s^{(\ell)}), \tag{15}$$



(a) (b)

Fig. 4. The densities $p_{\mathcal{C}(\mathcal{S})}(\mathbf{x})$ (a) and $p_{\mathcal{S}}(\mathbf{x})$ (b), which have been approximated using two iterations of component splitting and the subsequent mean-based partition.

Table I
Univariate Split Parameters for $R = 4, \lambda = 0.001$

j	$\tilde{w}^{(j)}$	$\tilde{m}^{(j)}$	$\tilde{\sigma}$
1	0.10766586425362	−1.42237156603631	0.58160633157686
2	0.39233413574638	−0.47412385534547	0.58160633157686
3	0.39233413574638	0.47412385534547	0.58160633157686
4	0.10766586425362	1.42237156603631	0.58160633157686

where

$$\boldsymbol{V}_s^{(\ell)} = [\boldsymbol{v}_{s,1}^{(\ell)} \quad \cdots \quad \boldsymbol{v}_{s,n_s}^{(\ell)}], \tag{16}$$

$$(\boldsymbol{\Lambda}_s^{(\ell)})^{-1/2} = \operatorname{diag}\left(\begin{bmatrix} \frac{1}{\sqrt{\lambda_{s,1}^{(\ell)}}} & \cdots & \frac{1}{\sqrt{\lambda_{s,n_s}^{(\ell)}}} \end{bmatrix}\right), \tag{17}$$

and $\mathbf{m}_s^{(\ell)}$ is the n_s-element position portion of the full-state mean, and the columns of $\boldsymbol{V}_s^{(\ell)}$ are the normalized eigenvectors of the position-marginal covariance $\mathbf{P}_s^{(\ell)}$, with $\boldsymbol{v}_{s,i}^{(\ell)}$ corresponding to the ith eigenvalue $\lambda_{s,i}^{(\ell)}$. In the transformed space,

$$p(\mathbf{y}) = \mathcal{N}(\mathbf{y};\ \mathbf{0},\ \boldsymbol{I}). \tag{18}$$

Note that, in defining the transformation over $\mathbb{X}_s$, the same transformation can be applied to the FoV, such that

$$\mathcal{S}_y^{(\ell)} = \{\boldsymbol{h}^{(\ell)}(\mathbf{s};\ \mathbf{m}_s^{(\ell)}, \mathbf{P}_s^{(\ell)}) : \mathbf{s} \in \mathcal{S}\}. \tag{19}$$

In $\mathbb{Y}$, the Euclidean distances to boundary points of $\mathcal{S}_y^{(\ell)}$ can be interpreted as probabilistically normalized distances. In fact, the Euclidean distance of a point $\mathbf{y}$ from the origin in $\mathbb{Y}$ corresponds exactly to the Mahalanobis distance between the corresponding point $\mathbf{s}$ and the original position-marginal component.

D. Component Selection and Collocation Points

Components are selected for splitting if they have sufficient weight and significant statistical overlap of the FoV boundaries ($\partial\mathcal{S}$). For components of sufficient weight, the change of variables is applied to the FoV to obtain $\mathcal{S}_y^{(\ell)}$ per (19). The overlap of the original component on $\mathcal{S}$ is then equivalent to the overlap of the standard Gaussian distribution on $\mathcal{S}_y^{(\ell)}$, which is quantified using a grid of collocation points on $\mathbb{Y}$, as shown in Fig. 2(b).

Define the collocation point $\bar{\mathbf{y}}_{i_1,\ldots,i_{n_s}} \in \mathbb{Y}$ such that

$$\bar{\mathbf{y}}_{i_1,\cdots,i_{n_s}} \triangleq [\bar{y}_1(i_1) \ \ldots \ \bar{y}_{n_s}(i_{n_s})]^T, \quad (i_1, \ldots, i_{n_s}) \in G, \tag{20}$$

$$\bar{y}_j(l) = -\zeta + 2\zeta\left(\frac{l-1}{N_g - 1}\right), \qquad j \in \mathbb{N}_{n_s}, \tag{21}$$

$$G = \{(i_1, \ldots, i_{n_s}) : i_{(\cdot)} \in \mathbb{N}_{N_g},\ \|\mathbf{y}_{i_1,\ldots,i_{n_s}}\| \leq \zeta\}, \tag{22}$$

where ζ is a user-specified bound for the grid, G is the set of indices of points that are within ζ of the origin, and N_g is the upper bound of the number of points per dimension. An inclusion variable is defined as

$$d_{i_1,\ldots,i_{n_s}}^{(\ell)} \triangleq 1_{\mathcal{S}_y^{(\ell)}}(\bar{\mathbf{y}}_{i_1,\ldots,i_{n_s}}). \tag{23}$$

Inclusion and exclusion patterns across the grid can be examined by first establishing an arbitrary reference index $(i'_1, \ldots, i'_{n_s}) \in G$. With this, $\varrho_{\mathcal{S}_y^{(\ell)}} \in \{0, 1\}$ is established to mark total inclusion or total exclusion as

$$\varrho_{\mathcal{S}_y^{(\ell)}} = \prod_G \delta_{d^{(\ell)}_{i'_1,\ldots,i'_{n_s}}}(d_{i_1,\ldots,i_{n_s}}^{(\ell)}), \tag{24}$$

which is equal to unity if all grid points lie inside of $\mathcal{S}_y^{(\ell)}$ or all grid points lie outside of $\mathcal{S}_y^{(\ell)}$, and is zero otherwise. If either all or no points are included, then no splitting is required. Otherwise, the component is marked for splitting.

E. Position Split Direction

Rather than split a component along each of its principal directions, a more judicious selection can be made by limiting split operations to a single direction (per component) per recursion. Thus, by performing one split per component per recursion, the component selection criteria are re-evaluated, reducing the overall number of components generated. In the aforementioned two-dimensional example, only a subset of new components generated from the first split are selected for further splitting, as shown in Fig. 3(b).

The split direction is chosen based on the relative geometry of the FoV, and thus position vectors are of interest. Choosing the best position split direction is a challenging problem. A common approach is to split along the component's covariance eigenvector with the largest eigenvalue [33]. This strategy, however, does not consider the FoV geometry and thus may increase the mixture size without improvement to the FoV-partitioned densities (11) and (12). Reference [35] provides a more sophisticated split direction criterion based on the integral linearization errors along the covariance eigenvectors. However, in the case that the FoV does not intersect the eigenvectors, this criterion cannot distinguish the best split direction. Another approach [36] determines the split direction based on the Hessian of the underlying nonlinear transformation, evaluated at the component mean. However, for the transformations considered in this paper of the form $g(\mathbf{s}) = c \cdot 1_{\mathcal{S}}(\mathbf{s})$, where c is some arbitrary constant, the Hessian either vanishes (for $\mathbf{s} \notin \partial\mathcal{S}$) or is undefined (for $\mathbf{s} \in \partial\mathcal{S}$).

Ideally, splitting along the chosen direction should minimize the number of splits required in the next iteration as well as improve the accuracy of the partition approximation applied after the final iteration. The computational complexity of exhaustive optimization of the split direction would likely negate the computational efficiency of the overall algorithm. Instead, to minimize the number of splits required in the next iteration, the position split direction is chosen as the direction that is orthogonal to the most grid planes of consistent inclusion/exclusion. Introducing a convenience function $s_j^{(\ell)} : \mathbb{N}_{N_g} \mapsto \{0, 1\}$, the plane of constant $y_j = \bar{y}_j(l)$ is consistently inside or consistently outside if

$$s_j^{(\ell)}(l) = \prod_{G, i_j = l} \delta_{d^{(\ell)}_{i'_1,\ldots,i_j,\ldots,i'_{n_s}}}(d^{(\ell)}_{i_1,\ldots,i_j,\ldots,i_{n_s}}) \tag{25}$$

is equal to unity, where $i'_1, \ldots, i_j, \ldots, i'_{n_s}$ is an arbitrary index tuple in G satisfying $i_j = l$, to which inclusion consistency is compared (see Appendix A for a numerical example). The optimal position split direction is then given by the eigenvector $\boldsymbol{v}_{s,j^*}$, where the optimal eigenvector index is found as

$$j^* = \arg\max_j \left(\sum_{l=1}^{N_g} s_j^{(\ell)}(l) \right). \tag{26}$$

For notational simplicity, the implicit dependence of j^* on the component index ℓ is omitted. For example, referring back to the two-dimensional example and Fig. 2(b), there are more rows than columns that are consistently inside or outside the transformed FoV, and thus $j^* = 2$ is chosen as the desired position split direction index. In the case where multiple maxima exist, the eigenvector with the largest eigenvalue is selected, which corresponds to the direction of the largest variance among the maximizing eigenvectors.

F. Multivariate Split of Full-State Component

Gaussian splitting must be performed along the principal directions of the full-state covariance. The general multivariate split approximation, splitting along the kth eigenvector $\boldsymbol{v}_k^{(\ell)}$ is given by [34]

$$w^{(\ell)}\mathcal{N}(\mathbf{x};\ \mathbf{m}^{(\ell)},\ \mathbf{P}^{(\ell)}) \approx \sum_{j=1}^{R} w^{(\ell,j)}\mathcal{N}(\mathbf{x};\ \mathbf{m}^{(\ell,j)},\ \mathbf{P}^{(\ell,j)}), \tag{27}$$

where

$$w^{(\ell,j)} = \tilde{w}^{(j)} w^{(\ell)}, \tag{28}$$

$$\mathbf{m}^{(\ell,j)} = \mathbf{m}^{(\ell)} + \sqrt{\lambda_k^{(\ell)}}\,\tilde{m}^{(j)} \boldsymbol{v}_k^{(\ell)}, \tag{29}$$

$$\mathbf{P}^{(\ell,j)} = \boldsymbol{V}^{(\ell)} \boldsymbol{\Lambda}^{(\ell)} \boldsymbol{V}^{(\ell)T}, \tag{30}$$

$$\boldsymbol{\Lambda}^{(\ell)} = \operatorname{diag}\left([\lambda_1 \cdots \tilde{\sigma}^2 \lambda_k \cdots \lambda_{n_x}]\right), \tag{31}$$

and the optimal univariate split parameters $\tilde{w}^{(j)}$, $\tilde{m}^{(j)}$, and $\tilde{\sigma}$ are found from the pre-computed split library given the number of split components R and regularization parameter λ. In general, the position components of the full-state eigenvectors will not perfectly match the desired position split vector due to correlations between the states. Rather, the actual full-state split is performed along $\boldsymbol{v}_{k*}^{(\ell)}$, where the optimal eigenvector index is found according to

$$k^* = \arg\max_k \left| \left[\boldsymbol{v}_{s,j^*}^{(\ell)T}\ \ \mathbf{0}^T \right] \boldsymbol{v}_k^{(\ell)} \right| \tag{32}$$

where, without loss of generality, a specific state convention is assumed such that position states are first in element order.

G. Recursion and Role of Negative Information

The splitting procedure is applied recursively, as detailed in Algorithm 1. The recursion is terminated when no remaining components satisfy the criteria for splitting. Each recursion further refines the GM near the FoV bounds to improve the approximations of (11) and (12). However, because a Gaussian component's split approximation (27) does not perfectly replicate the original component, a small error is induced with each split. Given enough recursions, this error may become dominant. In the authors' experience, the recursion is terminated well before the cumulative split approximation error dominates.

Algorithm 1 split_for_fov($\{w^{(\ell)}, \mathbf{m}^{(\ell)}, \mathbf{P}^{(\ell)}\}_{\ell=1}^{L}$, $w_{\min}, \mathcal{S}, R, \lambda$)

---

split $\leftarrow \{\}$, no_split $\leftarrow \{\}$
if $L = 0$ **then**
    return split
end if
for $\ell = 1, \ldots, L$ **do**
    if $w^{(\ell)} < w_{\min}$ **then**
        add $\{w^{(\ell)}, \mathbf{m}^{(\ell)}, \mathbf{P}^{(\ell)}\}$ to no_split
        continue
    end if
    Compute $\mathcal{S}_y^{(\ell)}$ according to (19)
    if $\varrho_{\mathcal{S}_y^{(\ell)}} = 1$ **then**
        add $\{w^{(\ell)}, \mathbf{m}^{(\ell)}, \mathbf{P}^{(\ell)}\}$ to no_split
    else
        $j^* \leftarrow$ equation (26) , $k^* \leftarrow$ equation (32)
        $\{w^{(\ell,j)}, \mathbf{m}^{(\ell,j)}, \mathbf{P}^{(\ell,j)}\}_{j=1}^{R} \leftarrow$ equation (27) with $k = k^*$
        add $\{w^{(\ell,j)}, \mathbf{m}^{(\ell,j)}, \mathbf{P}^{(\ell,j)}\}_{j=1}^{R}$ to split
    end if
end for
split $\leftarrow$ split_for_fov(split, $w_{\min}, \mathcal{S}, R, \lambda$)
return split $\cup$ no_split

One of the many potential applications of the recursive algorithm presented in this section involves incorporating the evidence of nondetections, or negative information, in single- or multiobject filtering. To demonstrate this, a single-object filtering problem with a bounded square FoV is considered where, in three subsequent sensor reports, no object is detected. The

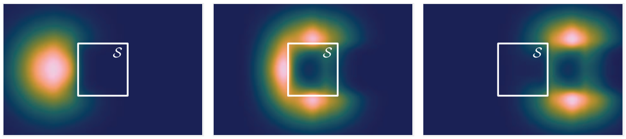


Fig. 5. Negative information, comprising absence of detections inside the sensor FoV $\mathcal{S}$, is used to update the object pdf as the object moves across the ROI.

true object position and constant velocity are unknown but are distributed according to a known GM pdf at the first time step. As the initial pdf is propagated over time, the position-marginal pdf travels from left to right, as shown in Fig. 5. For simplicity, the probability of detection inside the FoV is assumed equal to one. At each time step, the GM is refined by Algorithm 1 using $w_{\min} = 0.01$, $R = 3$, and $\lambda = 0.001$. Then, the mean-based partition approximation (12) is applied and the updated filtering density (9) is found. The results shown in Fig. 5 are obtained using a Matlab implementation of Algorithm 1. When executed on an Apple M1 Ultra processor with 64 GB RAM, the total execution time (over three time steps) of Algorithm 1 is 0.176 s, which translates to <60 ms per time step. As in many GM-based filters, the number of components may increase over time but can be reduced as needed through component merging and pruning.

H. Splitting for Multiple Regions

The presented splitting approach can be extended to accommodate multiple closed subsets, which may represent the FoVs in a multisensor network or imprecise measurements that take the form of multiple closed subsets, as is shown in Section V. For ease of exposition, the multiregion method is developed in the context of multiple FoVs with the awareness that the regions can be any bounded sets. Consider the case where the GM is to be partitioned about the boundaries of N_s FoVs $\{\mathcal{S}^{(\iota)}\}_{\iota=1}^{N_s}$. One simple approach to incorporate the multiple FoVs is to recursively apply Algorithm 1 for each FoV. Recall from Section IV-E, however, that the direction order in which components are split ultimately determines the total number of components generated. Thus, by the described naive approach, the resulting mixture size inherently depends on the order by which the FoVs are processed, which is undesirable.

Instead, the remainder of this subsection establishes a multi-FoV splitting algorithm that is invariant to FoV order. Given $\mathcal{S}^{(\iota)}$, denote by $\mathcal{S}_y^{(\iota,\ell)}$ the resulting transformed FoV for component ℓ via application of (19). Then, an inclusion variable similar to (23) is established as

$$d^{(\iota,\ell)}_{i_1,\ldots,i_{n_s}} \triangleq 1_{\mathcal{S}_y^{(\iota,\ell)}}(\bar{\mathbf{y}}_{i_1,\ldots,i_{n_s}}). \tag{33}$$

In each transformed FoV, grid points are either totally excluded or totally included if and only if

$$\varrho^{(\ell)}_{\{\mathcal{S}_y\}} = \prod_{\iota=1}^{N_s} \prod_{G} \delta_{d^{(\iota,\ell)}_{i'_1,\ldots,i'_{n_s}}} (d^{(\iota,\ell)}_{i_1,\ldots,i_{n_s}}) \tag{34}$$

is equal to unity, which indicates that a component does not require splitting. If a component is to be split, then the direction is chosen to minimize the ultimate mixture size, as discussed in Section IV-E. This is accomplished by identifying grid planes that are either consistently included/excluded in each FoV. Consistency of the plane of constant $y_j = \bar{y}_j(l)$ is indicated by

$$s_j^{(\ell)}(l) = \prod_{\iota=1}^{N_s} \prod_{G, i_j = l} \delta_{d^{(\iota,\ell)}_{i'_1,\ldots,i_j,\ldots,i'_{n_s}}} (d^{(\iota,\ell)}_{i_1,\ldots,i_j,\ldots,i_{n_s}}) \tag{35}$$

equal to unity. By this multi-FoV generalized indicator function, the optimal position split direction is found via (26). The complete multi-FoV splitting algorithm is summarized in Algorithm 2.

ALGORITHM 2
`split_for_multifov`$(\{w^{(\ell)}, \mathbf{m}^{(\ell)}, \mathbf{P}^{(\ell)}\}_{\ell=1}^{L}, w_{\min}, \{\mathcal{S}^{(\iota)}\}_{\iota=1}^{N_s}, R, \lambda)$

`split` $\leftarrow \{\}$, `no_split` $\leftarrow \{\}$
if $L = 0$ **then**
    return `split`
end if
for $\ell = 1, \ldots, L$ **do**
    if $w^{(\ell)} < w_{\min}$ **then**
        add $\{w^{(\ell)}, \mathbf{m}^{(\ell)}, \mathbf{P}^{(\ell)}\}$ to `no_split`
        continue
    end if
    for $\iota = 1, \ldots, N_s$ **do**
        compute $\mathcal{S}_y^{(\iota,\ell)}$ according to equation (19)
    end for
    if $\varrho^{(\ell)}_{\{\mathcal{S}_y\}} = 1$ **then**
        add $\{w^{(\ell)}, \mathbf{m}^{(\ell)}, \mathbf{P}^{(\ell)}\}$ to `no_split`
    else
        $j^* \leftarrow$ equation (26) , $k^* \leftarrow$ equation (32)
        $\{w^{(\ell,j)}, \mathbf{m}^{(\ell,j)}, \mathbf{P}^{(\ell,j)}\}_{j=1}^{R} \leftarrow$ equation (27) with $k = k^*$
        add $\{w^{(\ell,j)}, \mathbf{m}^{(\ell,j)}, \mathbf{P}^{(\ell,j)}\}_{j=1}^{R}$ to `split`
    end if
end for
`split` $\leftarrow$ `split_for_multifov(split,` $w_{\min}$, $\{\mathcal{S}^{(\iota)}\}_{\iota=1}^{N_s}, R, \lambda)$
return `split` $\cup$ `no_split`

The set inputs $\{\mathcal{S}^{(\iota)}\}$ in Algorithm 2 are not restricted to FoVs and can represent any regions. For example, two regions relevant to the human-robot interaction depicted in Fig. 1 are the human observer's binocular FoV and the tabletop region. The application of Algorithm

2 with respect to these two regions then enables the incorporation of the observation "The mug is on the table" in a GM Bayes filter, as is discussed in the following section.

V. APPLICATION TO IMPRECISE MEASUREMENTS

This section presents the application of the splitting algorithm to estimation problems involving imprecise measurements. Unlike traditional vector-type measurements, imprecise measurements are nonspecific, yet still contain valuable information. Examples of imprecise measurements include natural language statements [22], [23], inference rules [37, Sec. 22.2.4], and received signal strength type measurements under path-loss uncertainty [23], [38]. This section demonstrates the estimation of a person's location and velocity as they move through a public space using imprecise natural language measurements, as originally posed in [23]. Tracking is performed using a new GM Bernoulli filter for imprecise measurements, as discussed in the following subsections.

A. Imprecise Measurements

Imprecise measurements, such as those from natural language statements, can be modeled as RFSs and specified using *generalized likelihood functions*. For example, the statement

$$S = \text{“Felice is near the taco stand”} \tag{36}$$

provides some evidence about Felices's location, yet is not mutually exclusive[1] [1, p. 104, 126]. For simplicity, this paper adopts from [1, p. 105] the definition of being "near" a point $\mathbf{z}_0$ as belonging to a disc $\boldsymbol{\zeta} \subset \mathbb{Z}$ of radius D:

$$\boldsymbol{\zeta} = \{\mathbf{z} : \|\mathbf{z} - \mathbf{z}_0\| \leq D\}. \tag{37}$$

Although this specific natural language statement interpretation is considered for simplicity, the presented approach does not preclude more sophisticated models, such as in [22], [39]. The associated generalized likelihood function for this imprecise measurement is

$$\tilde{g}(\boldsymbol{\zeta}|\mathbf{x}) = P\{\mathbf{z} \in \boldsymbol{\zeta}\} = P\{\mathbf{h}(\mathbf{x}) \in \boldsymbol{\zeta}\}, \tag{38}$$

where $\mathbf{h} : \mathbb{X} \mapsto \mathbb{Z}$ is the deterministic mapping from the state space to the measurement space [23]. Generalized likelihood functions, such as those for natural language statements, are often nonlinear in $\mathbf{x}$. Through the presented Gaussian splitting approach and expansion of the nonlinear likelihood function about the GM component means, GM RFS filters can accommodate imprecise measurements, as demonstrated in the context of the RFS Bernoulli filter in the following subsection.

[1] In fact, this statement can further be considered vague or fuzzy due to uncertainty in the observer's definition of "near" [19, p. 266].

B. Bernoulli Filter for Imprecise Measurements

The Bernoulli filter is the Bayes-optimal filter for tracking a single object in the presence of false alarms, misdetections, and unknown object birth/death [1, Sec. 14]. A Bernoulli distribution is parameterized by a probability of object existence r and state pdf $p(\mathbf{x})$. The finite set statistics (FISST) density of a Bernoulli RFS is [1, p. 516]

$$f(X) = \begin{cases} 1 - r, & \text{if } X = \emptyset \\ r \cdot p(\mathbf{x}), & \text{if } X = \{\mathbf{x}\}. \end{cases} \tag{39}$$

Denote by p_b the conditional probability that the object is born given that it did not exist in the previous time step. Similarly, denote by p_S the conditional probability that the object survives to the next time step. The initial state of an object born at time k is assumed to be distributed according to the birth spatial density $b_k(\mathbf{x})$. Then, by the FISST generalized Chapman-Kolmogorov equation, the Bernoulli filter prediction equations are [1, p. 519]

$$\begin{aligned} p_{k|k-1}(\mathbf{x}) = & \frac{p_b \cdot (1 - r_{k-1|k-1}) b_{k|k-1}(\mathbf{x})}{r_{k|k-1}} \\ & + \frac{p_S r_{k-1|k-1} \int \pi_{k|k-1}(\mathbf{x}|\mathbf{x}') p_{k-1|k-1}(\mathbf{x}') \mathrm{d}\mathbf{x}'}{r_{k|k-1}}, \end{aligned} \tag{40}$$

$$r_{k|k-1} = p_b \cdot (1 - r_{k-1|k-1}) + p_S r_{k-1|k-1}, \tag{41}$$

where $\pi_{k|k-1}(\mathbf{x}|\mathbf{x}')$ is the single-object state transition density. Suppose that the spatial density and birth density are GMs and that the transition is linear-Gaussian:

$$p_{k-1|k-1}(\mathbf{x}) = \sum_{\ell=1}^{L_{k-1}} w_{k-1}^{(\ell)} \mathcal{N}(\mathbf{x};\ \mathbf{m}_{k-1}^{(\ell)},\ \mathbf{P}_{k-1}^{(\ell)}), \tag{42}$$

$$b_{k|k-1}(\mathbf{x}) = \sum_{\ell=1}^{L_{b,k}} \hat{w}_{b,k}^{(\ell)} \mathcal{N}(\mathbf{x};\ \mathbf{m}_{b,k}^{(\ell)},\ \mathbf{P}_{b,k}^{(\ell)}), \tag{43}$$

$$\pi_{k|k-1}(\mathbf{x}|\mathbf{x}') = \mathcal{N}(\mathbf{x};\ \mathbf{F}_{k-1}\mathbf{x}',\ \mathbf{Q}_{k-1}). \tag{44}$$

Then, the predicted spatial density at k is the sum of two GMs, given as

$$\begin{aligned} p_{k|k-1}(\mathbf{x}) = & \sum_{\ell=1}^{L_{b,k}} w_{b,k}^{(\ell)} \mathcal{N}(\mathbf{x};\ \mathbf{m}_{b,k}^{(\ell)},\ \mathbf{P}_{b,k}^{(\ell)}) \\ & + \sum_{\ell=1}^{L_{k-1}} w_{S,k|k-1}^{(\ell)} \mathcal{N}(\mathbf{x};\ \mathbf{m}_{S,k|k-1}^{(\ell)},\ \mathbf{P}_{S,k|k-1}^{(\ell)}), \end{aligned} \tag{45}$$

where

$$w_{b,k}^{(\ell)} = \hat{w}_{b,k}^{(\ell)} \frac{p_b \cdot (1 - r_{k-1|k-1})}{r_{k|k-1}}, \tag{46}$$

$$w_{S,k|k-1}^{(\ell)} = w_{k-1}^{(\ell)} \frac{p_S r_{k-1|k-1}}{r_{k|k-1}}, \tag{47}$$

$$\mathbf{m}_{S,k|k-1}^{(\ell)} = \mathbf{F}_{k-1} \mathbf{m}_{k-1}^{(\ell)}, \tag{48}$$

$$\mathbf{P}_{S,k|k-1}^{(\ell)} = \mathbf{F}_{k-1} \mathbf{P}_{k-1}^{(\ell)} \mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1}. \tag{49}$$

The predicted spatial density (45) can thus be expressed as a combined GM of the form

$$p_{k|k-1}(\mathbf{x}) = \sum_{\ell=1}^{L_{k|k-1}} w_{k|k-1}^{(\ell)} \mathcal{N}(\mathbf{x};\ \mathbf{m}_{k|k-1}^{(\ell)},\ \mathbf{P}_{k|k-1}^{(\ell)}), \tag{50}$$

where $\sum_{\ell=1}^{L_{k|k-1}} w_{k|k-1}^{(\ell)} = 1$.

The FoV-dependent probability of detection function is given by

$$p_D(\mathbf{x}; \mathcal{S}_k) = 1_{\mathcal{S}_k}(\mathbf{x}) p_D(\mathbf{s}), \tag{51}$$

where the single-argument function $p_D(\mathbf{s})$ is the corresponding probability of detection for an unbounded FoV. The measurement Υ_k is then a finite set

$$\Upsilon_k = \{\boldsymbol{\zeta}_1,\ \ldots,\ \boldsymbol{\zeta}_{m_k}\} \in \mathcal{F}(\mathfrak{Z}) \tag{52}$$

comprised of false alarms and a potentially empty imprecise measurement due a true object, where $\mathfrak{Z}$ is the set of all closed subsets of $\mathbb{Z}$ and $\mathcal{F}(\mathfrak{Z})$ is the space of all finite subsets of $\mathfrak{Z}$, as shown in [1, Ch. 5]. Assume that false alarms are Poisson distributed with rate λ_c and spatial density $\tilde{c}(\boldsymbol{\zeta})$. Then, the posterior state density and probability of existence are given by

$$p_{k|k}(\mathbf{x}) = \frac{1 - p_D(\mathbf{x}; \mathcal{S}_k) + p_D(\mathbf{x}; \mathcal{S}_k) \sum_{\boldsymbol{\zeta} \in \Upsilon_k} \frac{\tilde{g}_k(\boldsymbol{\zeta}|\mathbf{x})}{\lambda_c \tilde{c}(\boldsymbol{\zeta})}}{1 - \Delta_k} p_{k|k-1}(\mathbf{x}), \tag{53}$$

$$r_{k|k} = \frac{1 - \Delta_k}{1 - r_{k|k-1}\Delta_k} r_{k|k-1}, \tag{54}$$

where

$$\Delta_k = \int p_D(\mathbf{x}; \mathcal{S}_k) p_{k|k-1}(\mathbf{x}) \mathrm{d}\mathbf{x} - \sum_{\boldsymbol{\zeta} \in \Upsilon_k} \frac{\int p_D(\mathbf{x}; \mathcal{S}_k) \tilde{g}_k(\boldsymbol{\zeta}|\mathbf{x}) p_{k|k-1}(\mathbf{x}) \mathrm{d}\mathbf{x}}{\lambda_c \tilde{c}(\boldsymbol{\zeta})}, \tag{55}$$

which is a generalization of the result shown in [20] for state-dependent probability of detection.

Because (53) involves products of indicator functions and GMs, the resulting posterior density will not be a GM in general. Instead, the state-dependent probability of detection and generalized likelihood function can be expanded about the GM component means (see Appendix B), giving

$$p_{k|k}(\mathbf{x}) = \sum_{\ell=1}^{L_{k|k}} w_{k|k}^{(\ell)} \mathcal{N}(\mathbf{x};\ \mathbf{m}_{k|k}^{(\ell)},\ \mathbf{P}_{k|k}^{(\ell)}), \tag{56}$$

$$w_{k|k}^{(\ell)} = \frac{w_{k|k-1}^{(\ell)}}{1 - \Delta_k} \Bigg(1 - p_D(\mathbf{m}_{k|k-1}^{(\ell)}; \mathcal{S}_k) + p_D(\mathbf{m}_{k|k-1}^{(\ell)}; \mathcal{S}_k) \sum_{\boldsymbol{\zeta} \in \Upsilon_k} \frac{\tilde{g}_k(\boldsymbol{\zeta}|\mathbf{m}_{k|k-1}^{(\ell)})}{\lambda_c \tilde{c}(\boldsymbol{\zeta})} \Bigg), \tag{57}$$

$$\Delta_k = \sum_{\ell=1}^{L_{k|k-1}} w_{k|k-1}^{(\ell)} p_D(\mathbf{m}_{k|k-1}^{(\ell)}; \mathcal{S}_k) - \sum_{\boldsymbol{\zeta} \in \Upsilon_k} \frac{\sum_{\ell=1}^{L_{k|k-1}} w_{k|k-1}^{(\ell)} p_D(\mathbf{m}_{k|k-1}^{(\ell)}; \mathcal{S}_k) \tilde{g}_k(\boldsymbol{\zeta}|\mathbf{m}_{k|k-1}^{(\ell)})}{\lambda_c \tilde{c}(\boldsymbol{\zeta})} \tag{58}$$

$$\mathbf{m}_{k|k}^{(\ell)} = \mathbf{m}_{k|k-1}^{(\ell)}, \tag{59}$$

$$\mathbf{P}_{k|k}^{(\ell)} = \mathbf{P}_{k|k-1}^{(\ell)}. \tag{60}$$

The approximation error due to the zeroth-order expansion in (57) and (58) depends on the GM resolution near points of strong nonlinearity. In a high resolution mixture containing many components with small covariance matrices, the region about each mean in which the local approximation must be valid is correspondingly smaller compared to a low-resolution mixture [40]. Therefore, the recursive splitting method is employed to refine the mixture in nonlinear regions—specifically around $\partial\mathcal{S}_k$ and $\partial\boldsymbol{\zeta}_{(\cdot)}$—before computing the posterior GM (56). Then, the resulting posterior GM is reduced using one of many available algorithms for GM reduction [41]–[44]. This process, referred to as the GM Bernoulli filter for imprecise measurements, is summarized in Algorithm 3.

C. Airport Tracking Example

The recursive splitting approach is demonstrated in the context of tracking a person of interest through a crowded airport. This problem was originally posed in [23] and solved using a particle filter (PF) implementation of the Bernoulli filter. The object state is defined as

$$\mathbf{x}_k^T = [x_k \quad y_k \quad \dot{x}_k \quad \dot{y}_k] = [\mathbf{s}_k^T \quad \mathbf{v}_k^T], \tag{61}$$

where dimensionless distance units are used throughout. Measurements of the object are composed of natural language statements describing the person's current location in the form $Z_k = \{\zeta_{k,1}, \ldots, \zeta_{k,m_k}\}$, where m_k is the number of statements received at time k and

$$\zeta = a \Rightarrow \text{the object is near the anchor } a. \tag{62}$$

In (62), the integer $a \in \mathbb{A} \subset \mathbb{N}$ represents a fixed anchor, such as a taco stand or coffee shop, with corresponding known position $\mathbf{r}_a \in \mathbb{Z}$. Observers sometimes report incorrect statements (as false alarms) and sometimes fail to report true statements (as misdetections). The corresponding generalized likelihood function is

$$\tilde{g}_k(\zeta = a \,|\, \mathbf{x}_k) = \begin{cases} 1 & \text{if } \|\mathbf{s}_k - \mathbf{r}_a\| \leq 2d_a/3 \\ 0 & \text{otherwise} \end{cases}, \tag{63}$$

where d_a is the distance between anchor a and its nearest neighboring anchor. If the object is within $2d_a/3$ of anchor a, then the natural language statement reports that the object is near a (unless misdetected). Defining the

Algorithm 3 GM Bernoulli Filter for Imprecise Measurements

given $r_{0|0}$, $p_{0|0}(\mathbf{x})$

for $k = 1, \ldots, K$ **do**

Compute $r_{k|k-1}$ according to (41)

Compute $\{w^{(\ell)}_{S,k|k-1}, \mathbf{m}^{(\ell)}_{S,k|k-1}, \mathbf{P}^{(\ell)}_{S,k|k-1}\}_{\ell=1}^{L_{k|k-1}}$ according to (47)–(49)

Compute $\{w^{(\ell)}_{b,k}\}_{\ell=1}^{L_{b,k}}$ according to (46)

$\{w^{(\ell)}_{k|k-1}, \mathbf{m}^{(\ell)}_{k|k-1}, \mathbf{P}^{(\ell)}_{k|k-1}\}_{\ell=1}^{L_{k|k-1}} \leftarrow \{w^{(\ell)}_{S,k|k-1}, \mathbf{m}^{(\ell)}_{S,k|k-1}, \mathbf{P}^{(\ell)}_{S,k|k-1}\}_{\ell=1}^{L_{k-1}} \cup \{w^{(\ell)}_{b,k|k-1}, \mathbf{m}^{(\ell)}_{b,k|k-1}, \mathbf{P}^{(\ell)}_{b,k|k-1}\}_{\ell=1}^{L_{b,k}}$

$\{w^{(\ell)}_{k|k-1}, \mathbf{m}^{(\ell)}_{k|k-1}, \mathbf{P}^{(\ell)}_{k|k-1}\}_{\ell=1}^{L_{k|k-1}} \leftarrow$split_for_multifov$(\{w^{(\ell)}_{k|k-1}, \mathbf{m}^{(\ell)}_{k|k-1}, \mathbf{P}^{(\ell)}_{k|k-1}\}_{\ell=1}^{L_{k|k-1}}, w_{\min}, \{\mathcal{S}_k\} \cup \Upsilon_k, R, \lambda)$

Compute Δ_k according to (58)

Compute $r_{k|k}$ according to (54)

Compute $\{w^{(\ell)}_{k|k}, \mathbf{m}^{(\ell)}_{k|k}, \mathbf{P}^{(\ell)}_{k|k}\}_{\ell=1}^{L_{k|k}}$ according to (57),(59),(60)

$\{w^{(\ell)}_{k|k}, \mathbf{m}^{(\ell)}_{k|k}, \mathbf{P}^{(\ell)}_{k|k}\}_{\ell=1}^{L_{k|k}} \leftarrow$ reduce$(\{w^{(\ell)}_{k|k}, \mathbf{m}^{(\ell)}_{k|k}, \mathbf{P}^{(\ell)}_{k|k}\}_{\ell=1}^{L_{k|k}})$

end for

compact subset

$$\mathcal{A}_a = \{\mathbf{s} \,:\, \|\mathbf{s} - \mathbf{r}_a\| \le 2d_a/3\}\,, \tag{64}$$

the generalized likelihood function (63) can be written in terms of an indicator function as

$$\tilde{g}_k(\zeta = a \,|\, \mathbf{x}_k) = 1_{\mathcal{A}_a}(\mathbf{s}_k). \tag{65}$$

By this likelihood function, (57) and (58) simplify to

$$w^{(\ell)}_{k|k} = \frac{w^{(\ell)}_{k|k-1}}{1-\Delta_k}\Bigg(1 - p_D(\mathbf{m}^{(\ell)}_{k|k-1}; \mathcal{S}_k) + p_D(\mathbf{m}^{(\ell)}_{k|k-1}; \mathcal{S}_k)\sum_{\zeta \in Z_k} \frac{1_{\mathcal{A}_\zeta}(\mathbf{m}^{(\ell)}_{s,k|k-1})}{\lambda_c \tilde{c}(\zeta)}\Bigg), \tag{66}$$

$$\Delta_k = \sum_{\ell=1}^{L_{k|k-1}} w^{(\ell)}_{k|k-1} p_D(\mathbf{m}^{(\ell)}_{k|k-1}; \mathcal{S}_k) - \sum_{\zeta \in Z_k} \frac{\sum_{\ell=1}^{L_{k|k-1}} w^{(\ell)}_{k|k-1} p_D(\mathbf{m}^{(\ell)}_{k|k-1}; \mathcal{S}_k) 1_{\mathcal{A}_\zeta}(\mathbf{m}^{(\ell)}_{s,k|k-1}; \mathcal{S}_k)}{\lambda_c \tilde{c}(\zeta)}, \tag{67}$$

where λ_c denotes the clutter cardinality mean and the density of clutter $\tilde{c}(\zeta)$ is taken to be uniform over support $\mathbb{A}$.

The anchor locations and bounds $\partial\mathcal{A}_a$ are shown in Fig. 6. The gray shaded regions indicate exclusion regions the person cannot occupy due to physical barriers, and thus, $p_k(\mathbf{x}) = 0$ in these regions. Detections are reported every $T_k = 15\,[\mathrm{s}]$ and include an average of $\lambda_c = 0.25$ false detections. True detections are reported with a probability of detection $p_D(\mathbf{x}_k; \mathcal{S}_k)$ given by (51) with $p_D(\mathbf{s}_k) = 0.9$ and composite detection FoV

$$\mathcal{S}_k = \bigcup_{a \in \mathbb{A}} \mathcal{A}_a. \tag{68}$$

The object state is governed by the transition density

$$\pi_{k|k-1}(\mathbf{x}|\mathbf{x}') = \mathcal{N}(\mathbf{x};\ \mathbf{F}_{k-1}\mathbf{x}',\ \mathbf{Q}_{k-1}), \tag{69}$$

where

$$\mathbf{F}_k = \begin{bmatrix} 1 & 0 & T_k & 0 \\ 0 & 1 & 0 & T_k \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{70}$$

$$\mathbf{Q}_k = \begin{bmatrix} \frac{\varpi T_k^3}{3} & 0 & \frac{\varpi T_k^2}{2} & 0 \\ 0 & \frac{\varpi T_k^3}{3} & 0 & \frac{\varpi T_k^2}{2} \\ \frac{\varpi T_k^2}{2} & 0 & \varpi T_k & 0 \\ 0 & \frac{\varpi T_k^2}{2} & 0 & \varpi T_k \end{bmatrix}, \tag{71}$$

and $\varpi = 0.004$ is the intensity of the process noise.

The simulated reports are processed by the GM Bernoulli for imprecise measurements (Algorithm 3) and the Bernoulli PF [23] at each time step to obtain the posterior probability of existence and state density.

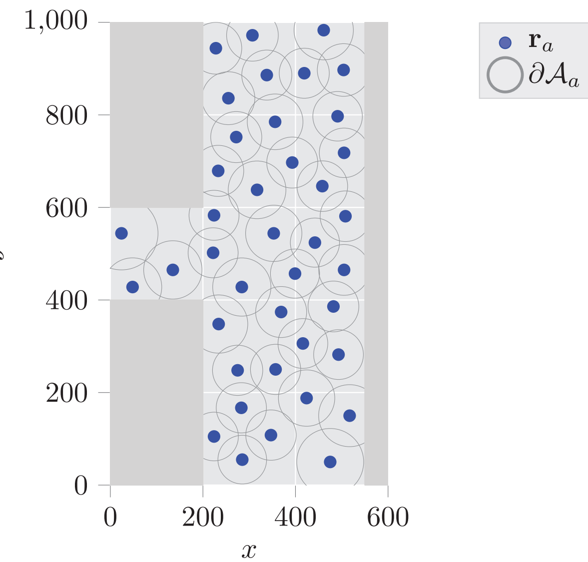


Fig. 6. Anchor locations and association extents.

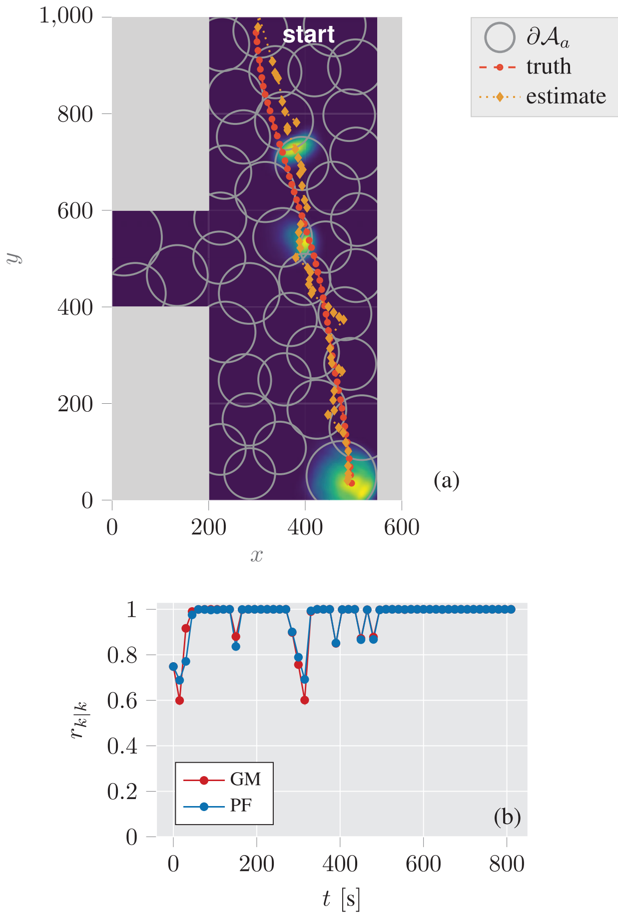


Fig. 7. (a) True trajectory and GM Bernoulli filter state estimates over time, where position state densities are shown for time steps $k = 15, 25, 55$ ($t = 225, 375, 825$ [s]), and (b) posterior probability of existence over time.

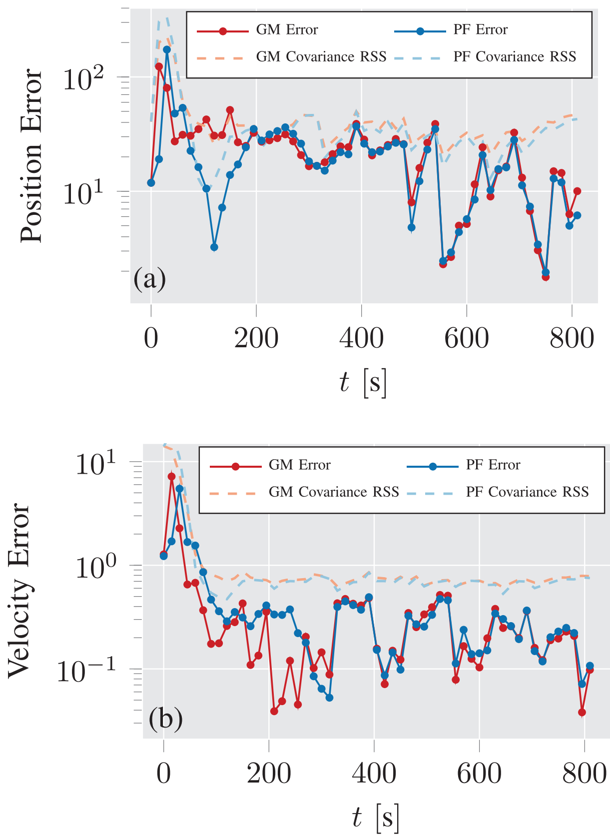


Fig. 8. MMSE estimation error and conditional covariance RSS of position (a) and velocity (b) states.

The Bernoulli PF is implemented using 5000 particles and a Markov chain Monte Carlo (MCMC) move step to improve sample diversity, as described in [23]. By splitting the density about the relevant anchor boundaries, the imprecise measurements are incorporated to refine the probabilistic belief and estimate the person's trajectory over time. The true trajectory, minimum mean square error (MMSE) estimates, and densities at select time steps are shown in Fig. 7(a). The Bernoulli PF estimates and densities are omitted for clarity. As shown, the true trajectory is consistently within the spatial distribution support.

The posterior probability of existence is shown over time in Fig. 7(b). The probability of existence of the object is consistently near one, falling momentarily to $r_{k|k} = 0.6$. This drop in probability appropriately reflects the increased uncertainty after three consecutive misdetections (the latter two of which are due to the object traveling outside detection bounds). As shown, the GM and PF approximations produce similar probability of existence estimates, where only slight differences are observed at times of nondetection.

The GM Bernoulli filter for imprecise measurements is exceptionally computationally efficient, resulting in a total simulation time of 45.2 s. When applied to identical measurement data, the Bernoulli PF simulation required 128.5 s. In fact, the largest computational bottleneck of the presented GM approach is the GM reduction step. A two-pass reduction strategy was found to effectively balance computational cost and estimation accuracy. The Mahalanobis distance-based merge strategy of [31] quickly reduces the number of GM components in the first pass. Then, if needed, the Kullback–Leibler divergence (KLD)-based Runnals algorithm [42] further reduces the mixture size to $L_{\max} = 100$.

The state estimation performance is quantified using the MMSE estimate error and the root sum squared (RSS) of the posterior conditional covariance, as shown in Fig. 8. The estimation performance of the GM filter is very similar to the Bernoulli PF, with neither method exhibiting a clear advantage in terms of estimation accuracy. The velocity root-sum-square (RSS) quickly converges to a steady state of approximately 0.7 [dist/s], the lower bound of which is largely determined by the person's assumed maneuverability and associated process noise covariance. Similarly, the largest uncertainty is observed near $k = 21$ ($t = 315$ [s]), after three consecutive misdetections.

While this example considers single-object estimation, the expansion approximation and splitting approach described in Section V-B are applicable to any GM RFS filter and thus can be used in multiobject estimation problems. In the presented example of tracking a person of interest and its multiobject extension involving multiple persons of interest, the posterior RFS density can be used to intelligently query or deploy resources to find or intercept persons of interest. In this case, one particularly useful statistic is the probability that a given number of individuals are near a particular anchor. This information is fully described by the RFS FoV cardinality distribution, as presented in the following section.

VI. FOV CARDINALITY DISTRIBUTION

This section presents pmfs for the cardinality of objects inside a bounded FoV $\mathcal{S}$ given different global multiobject densities $f(\cdot)$. Previous work derived expressions for the first and second moments of FoV cardinality distributions given Poisson, independently and identically distributed cluster (i.i.d.c.) [45], and multi-Bernoulli (MB) [46] global densities. This section instead develops full pmfs expressions, from which first, second, or any higher-order moments can be easily obtained [47, Ch. 30]. A similar concept is discussed in [37] in the context of "censored" RFSs, and a general expression is provided in terms of set derivatives and belief mass functions. This paper presents a new direct approach to obtain FoV cardinality distributions based on conditional cardinality functions and derives new simplified expressions for representative RFS distribution classes. The Poisson, i.i.d.c., MB, and generalized labeled multi-Bernoulli (GLMB) distributions are considered in Sections VI-A, VI-B, VI-C, and VI-D, respectively.

The probability of n objects existing inside FoV $\mathcal{S}$ conditioned on X can be written in terms of the indicator function as

$$\rho_{\mathcal{S}}(n \mid X) = \sum_{X^n \subseteq X} [1_{\mathcal{S}}(\cdot)]^{X^n} [1 - 1_{\mathcal{S}}(\cdot)]^{X \setminus X^n}, \tag{72}$$

where the summation is taken over all subsets $X^n \subseteq X$ with cardinality n. Given the RFS density $f(X)$, the FoV cardinality distribution is obtained via the set integral as

$$\rho_{\mathcal{S}}(n) = \int \rho_{\mathcal{S}}(n \mid X) f(X) \delta X. \tag{73}$$

Expanding the integral,

$$\rho_{\mathcal{S}}(n) = \sum_{m=n}^{\infty} \frac{1}{m!} \int_{\mathbb{X}^m} \rho_{\mathcal{S}}(n \mid \{\mathbf{x}_1, ..., \mathbf{x}_m\}) f(\{\mathbf{x}_1, ..., \mathbf{x}_m\}) \mathrm{d}\mathbf{x}_1 \cdots \mathrm{d}\mathbf{x}_m. \tag{74}$$

Remark: The results presented in this section can be trivially extended to express the predicted cardinality of object-originated *detections* Z (excluding false alarms) by noting that

$$\rho_{\mathcal{S}}(n_Z \mid X) = \sum_{X^n \subseteq X} [p_D(\cdot) 1_{\mathcal{S}}(\cdot)]^{X^n} [1 - p_D(\cdot) 1_{\mathcal{S}}(\cdot)]^{X \setminus X^n}, \tag{75}$$

where $n_Z = |Z|$.

A. Poisson Distribution

The density of a Poisson-distributed RFS is

$$f(X) = e^{-N_X} [D]^X, \tag{76}$$

where N_X is the global cardinality mean and $D(\mathbf{x})$ is the probability hypothesis density (PHD), or intensity function, of X, which is defined on the single-object space $\mathbb{X}$. One important property of the PHD is that its integral over a closed set on $\mathbb{X}$ yields the expected number of objects within that set, i.e.,

$$E[|X \cap T|] = \int_T D(\mathbf{x}) \mathrm{d}\mathbf{x}. \tag{77}$$

Proposition 1 *Given a Poisson-distributed RFS with PHD $D(\mathbf{x})$ and global cardinality mean N_X, the cardinality of objects inside the FoV $\mathcal{S} \subseteq \mathbb{X}_s$ is distributed according to*

$$\rho_{\mathcal{S}}(n) = \sum_{m=n}^{\infty} \frac{e^{-N_X}}{n!(m-n)!} \langle 1_{\mathcal{S}}, D \rangle^n \langle 1 - 1_{\mathcal{S}}, D \rangle^{m-n}. \tag{78}$$

Proof: Substituting (76) into (74), we get

$$\rho_{\mathcal{S}}(n) = \sum_{m=n}^{\infty} \frac{1}{m!} e^{-N_X} \int_{\mathbb{X}^m} \sum_{X^n \subseteq X} [1_{\mathcal{S}}(\cdot) D(\cdot)]^{X^n} \cdot [(1 - 1_{\mathcal{S}}(\cdot)) D(\cdot)]^{X \setminus X^n} \mathrm{d}\mathbf{x}_1 \cdots \mathrm{d}\mathbf{x}_{m.} \tag{79}$$

The nested integrals of (79) can be distributed, rewriting the second sum over n-cardinality index sets $\mathcal{I}^n$ as

$$\rho_{\mathcal{S}}(n) = \sum_{m=n}^{\infty} \frac{1}{m!} e^{-N_X} \sum_{\mathcal{I}^n \subseteq \mathbb{N}_m} \left[\int 1_{\mathcal{S}}(\mathbf{x}_{(\cdot)}) D(\mathbf{x}_{(\cdot)}) \mathrm{d}\mathbf{x}_{(\cdot)} \right]^{\mathcal{I}^n} \cdot \left[\int (1 - 1_{\mathcal{S}}(\mathbf{x}_{(\cdot)})) D(\mathbf{x}_{(\cdot)}) \right]^{\mathbb{N}_m \setminus \mathcal{I}^n}. \tag{80}$$

Note that the value of the integrals is independent of the product index i, and thus

$$\rho_{\mathcal{S}}(n) = \sum_{m=n}^{\infty} e^{-N_X} \frac{1}{m!} \frac{m!}{n!(m-n)!} \langle 1_{\mathcal{S}}, D \rangle^n \langle 1 - 1_{\mathcal{S}}, D \rangle^{m-n}, \tag{81}$$

from which (78) follows. □

Remark: Computation of (78) requires only one integral computation, namely $\langle 1_{\mathcal{S}}, D \rangle$, which can be found either by summing the weights of (11) or through Monte Carlo integration. Using the integral property of the PHD (77), the integral

$$\langle 1 - 1_{\mathcal{S}}, D \rangle = N_X - \langle 1_{\mathcal{S}}, D \rangle. \tag{82}$$

Furthermore, for $m \gg N_X$, the summand of (78) is negligible, and the infinite sum can be safely truncated at an appropriately chosen $m = m_{\max}(N_X)$.

B. Independent Identically Distributed Cluster Distribution

The density of an i.i.d.c. RFS is

$$f(X) = |X|! \cdot \rho(|X|)[p]^X\,, \tag{83}$$

where $\rho(n)$ is the cardinality pmf and $p(\mathbf{x})$ is the single-object state pdf.

Proposition 2 *Given an i.i.d.c.-distributed RFS with cardinality pmf $\rho(\cdot)$ and state density $p(\cdot)$, the cardinality of objects inside the FoV $\mathcal{S}$ is distributed according to*

$$\rho_{\mathcal{S}}(n) = \sum_{m=n}^{\infty} \rho(m) \binom{m}{n} \langle 1_{\mathcal{S}}, p\rangle^n \langle 1 - 1_{\mathcal{S}}, p\rangle^{m-n}, \tag{84}$$

where $\binom{m}{n}$ is the binomial coefficient.

Proof: Substituting (83) into (74), we get

$$\rho_{\mathcal{S}}(n) = \sum_{m=n}^{\infty} \frac{1}{m!} m! \rho(m) \tag{85}$$

$$\int\limits_{\mathbb{X}^m} \sum_{X^n \subseteq X} \cdot [1_{\mathcal{S}}(\cdot)p(\cdot)]^{X^n} [(1 - 1_{\mathcal{S}}(\cdot))p(\cdot)]^{X \setminus X^n} \mathrm{d}\mathbf{x}_1 \cdots \mathrm{d}\mathbf{x}_m.$$

The integral can be moved inside the products so that

$$\rho_{\mathcal{S}}(n) = \sum_{m=n}^{\infty} \rho(m) \sum_{\mathcal{I}^n \subseteq \mathbb{N}_m} \left[\int 1_{\mathcal{S}}(\mathbf{x}_{(\cdot)}) p(\mathbf{x}_{(\cdot)}) \mathrm{d}\mathbf{x}_{(\cdot)}\right]^{\mathcal{I}^n}$$
$$\cdot \left[\int (1 - 1_{\mathcal{S}}(\mathbf{x}_{(\cdot)})) p(\mathbf{x}_{(\cdot)}) \mathrm{d}\mathbf{x}_{(\cdot)}\right]^{\mathbb{N}_m \setminus \mathcal{I}^n}. \tag{86}$$

Equation (84) follows from (86) by noting that there are $\binom{m}{n}$ unique unordered n-cardinality index subsets of $\mathbb{N}_m$. □

C. MB Distribution

The density of a MB distribution is [37, p. 102]

$$f(X) = \left[\left(1 - r^{(\cdot)}\right)\right]^{\mathbb{N}_M} \sum_{1 \le i_1 \ne \cdots \ne i_n \le M} \left[\frac{r^{i_{(\cdot)}} p^{i_{(\cdot)}}(\mathbf{x}_{(\cdot)})}{1 - r^{i_{(\cdot)}}}\right]^{\mathbb{N}_n}, \tag{87}$$

where M is the number of MB components and maximum possible object cardinality, r^i is the probability that the ith object exists, and $p^i(\mathbf{x})$ is the single-object state density of the ith object if it exists.

Proposition 3 *Given at MB density of the form of (87), the cardinality of objects inside the FoV $\mathcal{S}$ is distributed according to*

$$\rho_{\mathcal{S}}(n) = \left[\left(1 - r^{(\cdot)}\right)\right]^{\mathbb{N}_M}$$
$$\cdot \sum_{\mathcal{I}_1 \uplus \mathcal{I}_2 \subseteq \mathbb{N}_M} \delta_n(|\mathcal{I}_1|) \left[\frac{\langle 1_{\mathcal{S}}, r^{(\cdot)} p^{(\cdot)}\rangle}{1 - r^{(\cdot)}}\right]^{\mathcal{I}_1} \left[\frac{\langle 1 - 1_{\mathcal{S}}, r^{(\cdot)} p^{(\cdot)}\rangle}{1 - r^{(\cdot)}}\right]^{\mathcal{I}_2}, \tag{88}$$

where the summation is taken over all mutually exclusive index partitions $\mathcal{I}_1, \mathcal{I}_2$ such that $\mathcal{I}_1 \cup \mathcal{I}_2 \subseteq \mathbb{N}_M$.

Proof of Proposition 3 is given in Appendix C. Within a given summand term of (88), the index sets $\mathcal{I}_1, \mathcal{I}_2$, and $\mathbb{N}_M \setminus (\mathcal{I}_1 \cup \mathcal{I}_2)$ can be interpreted as the indices of objects within the FoV, objects outside the FoV, and nonexistent objects, respectively. Following the same procedure, similar results for the labeled multi-Bernoulli (LMB) [3] and multi-Bernoulli mixture (MBM) [48] RFS distributions may be obtained.

Direct computation of (88) is only feasible for small M due to the sum over all permutations $\mathcal{I}_1 \uplus \mathcal{I}_2 \subseteq \mathbb{N}_M$. For large M, an alternative formulation based on Fourier transforms allows fast numerical computation. For each MB component, the integral $\langle 1_{\mathcal{S}}, p^{(i)}\rangle$ is computed either by summing the weights of the partitioned GM or by Monte Carlo integration. Using the integral results, the probability of object i existing inside the FoV is found as

$$r_{\mathcal{S}}^{(i)} = r^{(i)} \langle 1_{\mathcal{S}}, p^{(i)}\rangle. \tag{89}$$

Then, as shown in [49], (88) can be equivalently written as

$$\rho_{\mathcal{S}}(n) = \frac{1}{M+1} \times \tag{90}$$
$$\sum_{m=0}^{M} \left\{ e^{-j2\pi mn/(M+1)} \prod_{k=1}^{M} \left[r_{\mathcal{S}}^{(k)} e^{j2\pi m/(M+1)} + (1 - r_{\mathcal{S}}^{(k)})\right]\right\}$$

and solved using the discrete Fourier transform, for which a number of efficient algorithms exist.

D. GLMB Distribution

The density of a GLMB distribution is given by [2]

$$\mathring{f}(\mathring{X}) = \Delta(\mathring{X}) \sum_{\xi \in \Xi} w^{(\xi)}(\mathcal{L}(\mathring{X}))[p^{(\xi)}]^{\mathring{X}}\,, \tag{91}$$

where each $\xi \in \Xi$ represents a history of measurement association maps, each $p^{(\xi)}(\cdot, \ell)$ is a probability density on $\mathbb{X}$, and each weight $w^{(\xi)}$ is nonnegative with $\sum_{(I,\xi) \in \mathcal{F}(\mathbb{L}) \times \Xi} w^{(\xi)}(I) = 1$. The label of a labeled state $\mathring{x}$ is recovered by $\mathcal{L}(\mathring{x})$, where $\mathcal{L} : \mathbb{X} \times \mathbb{L} \mapsto \mathbb{L}$ is the projection defined by $\mathcal{L}((\mathbf{x}, \ell)) \triangleq \ell$. Similarly, for LRFSs, $\mathcal{L}(\mathring{X}) \triangleq \{\mathcal{L}(\mathring{x}) : \mathring{x} \in \mathring{X}\}$. The distinct label indicator $\Delta(\mathring{X}) = \delta_{(|\mathring{X}|)}(|\mathcal{L}(\mathring{X})|)$ ensures that only sets with distinct labels are considered.

Proposition 4 *Given a GLMB density $\mathring{f}(\mathring{X})$ of the form of (91), the cardinality of objects inside a bounded FoV $\mathcal{S}$ is distributed according to*

$$\rho_{\mathcal{S}}(n) = \sum_{(\xi,\mathcal{I}_1 \uplus \mathcal{I}_2)\in \Xi\times\mathcal{F}(\mathbb{L})} w^{(\xi)}(I)\delta_n(|\mathcal{I}_1|)\langle 1_{\mathcal{S}}, p\rangle^{\mathcal{I}_1}\langle 1-1_{\mathcal{S}}, p\rangle^{\mathcal{I}_2}. \quad (92)$$

Proof: Equation (72) can be rewritten to accommodate the labeled RFS as

$$\rho_{\mathcal{S}}(n\,|\,\mathring{X}) = \sum_{\mathring{X}^n\subseteq\mathring{X}} [1_{\mathcal{S}}(\cdot)]^{\mathring{X}^n}[1-1_{\mathcal{S}}(\cdot)]^{\mathring{X}\setminus\mathring{X}^n}. \quad (93)$$

If $\mathring{X}$ is distributed according to the LRFS density $\mathring{f}(\mathring{X})$, the FoV cardinality distribution is obtained via the set integral

$$\rho_{\mathcal{S}}(n) = \int \rho_{\mathcal{S}}(n\,|\,\mathring{X})\mathring{f}(\mathring{X})\delta\mathring{X}. \quad (94)$$

Expanding the integral,

$$\begin{aligned}\rho_{\mathcal{S}}(n) &= \sum_{m=n}^{\infty}\frac{1}{m!}\sum_{(\ell_1,\ldots,\ell_m)\in\mathbb{L}^m}\int_{\mathbb{X}^m}\rho_{\mathcal{S}}(n\,|\,\{(\mathbf{x}_1,\ell_1),\ldots,(\mathbf{x}_m,\ell_m)\})\\ &\quad\cdot\mathring{f}(\{(\mathbf{x}_1,\ell_1),\ldots,(\mathbf{x}_m,\ell_m)\})\mathrm{d}\mathbf{x}_1\cdots\mathrm{d}\mathbf{x}_m. \end{aligned}\quad (95)$$

Defining $p^{(\xi,\ell)}(x) \triangleq p^{(\xi)}(x,\ell)$, substitution of (91) and (93) yields

$$\begin{aligned}\rho_{\mathcal{S}}(n) &= \sum_{m=n}^{\infty}\frac{1}{m!}m!\sum_{\{\ell_1,\ldots,\ell_m\}\in\mathbb{L}^m}\sum_{\xi\in\Xi}w^{(\xi)}(\{\ell_1,\ldots,\ell_m\})\\ &\quad\sum_{I^n\subseteq\{\ell_1,\ldots\ell_m\}}\langle 1_{\mathcal{S}}, p^{(\xi,\cdot)}\rangle^{I^n}\langle 1-1_{\mathcal{S}}, p^{(\xi,\cdot)}\rangle^{\{\ell_1,\ldots,\ell_m\}\setminus I^n}\\ &= \sum_{(\xi,I)\in\Xi\times\mathcal{F}(\mathbb{L})}w^{(\xi)}(I)\sum_{I^n\subseteq I}\langle 1_{\mathcal{S}}, p^{(\xi,\cdot)}\rangle^{I^n}\langle 1-1_{\mathcal{S}}, p^{(\xi,\cdot)}\rangle^{I\setminus I^n},\end{aligned}\quad (96)$$

from which (92) follows. □

Remark: Substitution of $n = 0$ in (92) gives the GLMB void probability functional [6, Eq. (22)], which, while less general, has theoretical significance and practical applications in sensor management.

VII. SENSOR PLACEMENT EXAMPLE

The FoV statistics developed in this paper are demonstrated through a sensor placement optimization problem subject to multiobject uncertainty. The global distribution is assumed to be MB-distributed. Numerical simulation is performed for the case of 100 MB components, with probabilities of existence randomly chosen between 0.35 and 1. Each MB component has a Gaussian density and randomly chosen mean and covariance. To visualize the global distribution, the PHD is shown in Fig. 9.

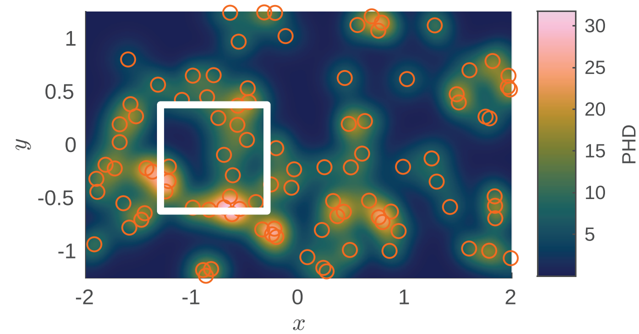


Fig. 9. PHD of the global MB distribution with 100 potential objects, where object means are represented by orange circles and the bounds of the FoV that maximize the FoV cardinality variance are shown in white.

The PHD is analogous to the expected value for RFSs and is defined as [50]

$$D(\mathbf{x}) \triangleq \mathrm{E}[\delta_X(\mathbf{x})] = \int \delta_X(\mathbf{x})\cdot f(X)\delta X, \quad (97)$$

for an arbitrary RFS X with density $f(X)$, where

$$\delta_X(\mathbf{x}) \triangleq \sum_{\mathbf{w}\in X}\delta_{\mathbf{w}}(\mathbf{x}). \quad (98)$$

It follows that the PHD of an MB RFS (87) is [37, p. 102]

$$D(\mathbf{x}) = \sum_{i=1}^{M} r^i p^i(\mathbf{x}). \quad (99)$$

The objective of the sensor control problem is to place the FoV, comprising a square of 1×1 dimensions, in the ROI (Fig. 9) such that the variance of object cardinality inside the FoV is maximized. This objective can be interpreted as placing the FoV in a region of the ROI where the object cardinality is most uncertain. A related objective that minimizes the variance of the *global* cardinality using CB-MeMBer predictions was first proposed in [5]. For each candidate FoV placement, the FoV cardinality pmf is given by (88) and efficiently computed using (90). The variance of the resulting pmf is shown as a function of the FoV center location in Fig. 10. The optimal FoV center location is found to be $(-0.8, -1.25)$.

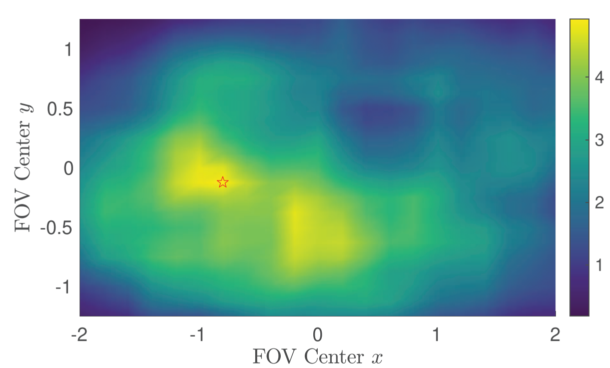


Fig. 10. FoV cardinality variance as a function of FoV center location, where the red star denotes the maximum variance point.

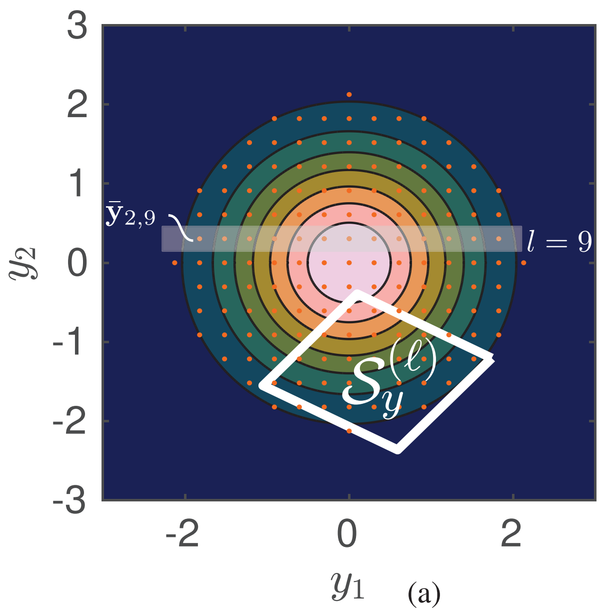


Fig. 11. (a) True trajectory and state estimates over time, where position state densities are shown for time steps $k = 15, 25, 55$ ($t = 225, 375, 825$ [s]), and (b) posterior probability of existence over time.

A compelling result is that, by virtue of the bounded FoV geometry, spatial information is encoded in the FoV cardinality pmf. It can be seen that the optimal FoV (Fig. 9) has boundary segments (lower half of left boundary and right half of lower boundary) that bisect clusters of MB components. These boundary segments divide the components' single-object densities such that significant mass appears inside and outside the FoV, increasing the overall FoV cardinality variance.

VIII. CONCLUSIONS

This paper presents an approach for incorporating bounded FoV geometry into state density updates and object cardinality predictions via FISST. Inclusion/exclusion evidence such as negative information and soft evidence is processed in state density updates via a novel Gaussian splitting algorithm that recursively refines a Gaussian mixture approximation near the boundaries of the discrete FoV geometry. Using FISST, cardinality pmfs that describe the probability that a given number of objects exist inside the FoV are derived. The approach is presented for representative labeled and unlabeled RFS distributions and, thus, is applicable to a wide range of tracking, perception, and sensor planning problems.

APPENDIX A Inclusion Consistency Example

Consider a plane of constant $y_2 = \bar{y}_2(9)$—that is, $j = 2$ and $l = 9$. As shown in Fig. 11, the index $l = 9$ denotes the ninth grid plane from the bottom. To evaluate inclusion/exclusion consistency in this plane, an arbitrary reference point is selected as $\bar{\mathbf{y}}_{2,9}$ (where the corresponding indices are $i_1' = 2$ and $i_j = i_2 = l = 9$). Note that this reference index tuple (2,9) belongs to G (depicted by the set of orange dots) and lies in the plane of constant $i_j = l$.

It is apparent from Fig. 11 that $\bar{\mathbf{y}}_{2,9} \notin \mathcal{S}_y^{(\ell)}$. Thus, the corresponding component inclusion variable (23) for the selected reference point is

$$d_{i_1',i_2}^{(\ell)} = d_{2,9}^{(\ell)} = 1_{\mathcal{S}_y^{(\ell)}}(\bar{\mathbf{y}}_{2,9}) = 0. \tag{100}$$

In the following inclusion/exclusion consistency check, which follows from (25), the inclusion variables are computed for all remaining points in the plane and compared to $d_{2,9}$:

$$\begin{aligned} s_j^{(\ell)}(l) = s_2^{(\ell)}(9) &= \prod_{G,i_2=9} \delta_{d_{2,9}^{(\ell)}}(d_{i_1,9}^{(\ell)}) \\ &= \delta_{d_{2,9}^{(\ell)}}(d_{2,9}^{(\ell)}) \cdot \delta_{d_{2,9}^{(\ell)}}(d_{3,9}^{(\ell)}) \cdots \delta_{d_{2,9}^{(\ell)}}(d_{14,9}^{(\ell)}) \\ &= \delta_0(0) \cdot \delta_0(0) \cdots \delta_0(0) = 1, \end{aligned} \tag{101}$$

where it is noted that i_1 ranges from 2 to 14 in the considered plane (in which there are thirteen corresponding orange dots). Thus, $s_j^{(\ell)}(l) = 1$ signifies that the plane is indeed consistently inside or consistently outside the FoV, the latter of which is easily verified by inspecting Fig. 11.

APPENDIX B Taylor Series Expansion About Means

Equation (53) can be written compactly as

$$p_{k|k}(\mathbf{x}) = \alpha(\mathbf{x}) p_{k|k-1}(\mathbf{x}) \tag{102}$$

$$= \sum_{\ell=1}^{L_{k|k-1}} \alpha(\mathbf{x}) w_{k|k-1}^{(\ell)} \mathcal{N}(\mathbf{x};\ \mathbf{m}_{k|k-1}^{(\ell)},\ \mathbf{P}_{k|k-1}^{(\ell)}), \tag{103}$$

where

$$\alpha(\mathbf{x}) = \frac{1 - p_D(\mathbf{x}; \mathcal{S}_k) + p_D(\mathbf{x}; \mathcal{S}_k) \sum_{\zeta \in \Upsilon_k} \frac{\tilde{g}_k(\zeta|\mathbf{x})}{\lambda_c \tilde{c}(\zeta)}}{1 - \Delta_k} \tag{104}$$

and where the functional dependence of α on the FoV and measurement is omitted for brevity. The function $\alpha(\mathbf{x})$ can be approximated locally by a Taylor series expansion about a given component mean as

$$\alpha(\mathbf{x}) \approx \alpha(\mathbf{m}_{k|k-1}^{(\ell)}) + \left(\frac{\partial \alpha}{\partial \mathbf{x}}\right)\bigg|_{\mathbf{x}=\mathbf{m}_{k|k-1}^{(\ell)}} (\mathbf{x} - \mathbf{m}_{k|k-1}^{(\ell)}) + \cdots. \tag{105}$$

To zeroth order, $\alpha(\mathbf{x}) \approx \alpha(\mathbf{m}_{k|k-1}^{(\ell)})$, such that

$$p_{k|k}(\mathbf{x}) \approx \sum_{\ell=1}^{L_{k|k-1}} \alpha(\mathbf{m}_{k|k-1}^{(\ell)}) w_{k|k-1}^{(\ell)} \mathcal{N}(\mathbf{x};\ \mathbf{m}_{k|k-1}^{(\ell)},\ \mathbf{P}_{k|k-1}^{(\ell)}), \tag{106}$$

from which (56)–(60) follow.

APPENDIX C Proof of Proposition 3

Let $\mathbb{K}_M^{(n)} \triangleq \{(i_1, ..., i_n) : 1 \le i_1 \ne \cdots \ne i_n \le M\}$. Then, (87) can be rewritten as

$$f(X) = \left[\left(1 - r^{(\cdot)}\right)\right]^{\mathbb{N}_M} \sum_{(\mathcal{I}_\sigma) \in \mathbb{K}_M^{(n)}} \left[\frac{r^{i_{(\cdot)}} p^{i_{(\cdot)}}(x_{(\cdot)})}{1 - r^{i_{(\cdot)}}}\right]^{\mathbb{N}_n}, \tag{107}$$

where $\mathcal{I}_\sigma$ denotes the (unordered) set $\{i_1, ..., i_n\}$ and $(\mathcal{I}_\sigma)$ denotes the (ordered) sequence $(i_1, ..., i_n) = (\alpha_{\sigma(1)}, ..., \alpha_{\sigma(n)})$, where the n-tuple index set $\{\alpha_1, ..., \alpha_n\} \subseteq \mathbb{N}_M$ and σ is a permutation of $\mathbb{N}_n$.

Substituting (107) into (74),

$$\begin{aligned}\rho_{\mathcal{S}}(n) &= \left[\left(1 - r^{(\cdot)}\right)\right]^{\mathbb{N}_M} \\ &\cdot \sum_{m=n}^{M} \frac{1}{m!} \int_{\mathbb{X}^m} \sum_{(\mathcal{I}_\sigma) \in \mathbb{K}_M^{(n)}} \delta_m(|\mathcal{I}_\sigma|) \left[\frac{r^{i_{(\cdot)}} p^{i_{(\cdot)}}(\mathbf{x}_{(\cdot)})}{1 - r^{i_{(\cdot)}}}\right]^{\mathbb{N}_m} \\ &\cdot \sum_{X^n \subseteq X} [1_{\mathcal{S}}(\cdot)]^{X^n} [1 - 1_{\mathcal{S}}(\cdot)]^{X \setminus X^n} \mathrm{d}\mathbf{x}_1 \cdots \mathrm{d}\mathbf{x}_m. \end{aligned} \tag{108}$$

The last sum can be written in terms of label index sets $\mathcal{I}_1 \uplus \mathcal{I}_2 = \mathcal{I}_\sigma$ as

$$\begin{aligned}\rho_{\mathcal{S}}(n) &= \left[\left(1 - r^{(\cdot)}\right)\right]^{\mathbb{N}_M} \\ &\cdot \sum_{m=n}^{M} \frac{1}{m!} \int_{\mathbb{X}^m} \sum_{(\mathcal{I}_\sigma) \in \mathbb{K}_M^{(n)}} \delta_m(|\mathcal{I}_\sigma|) \left[\frac{r^{i_{(\cdot)}} p^{i_{(\cdot)}}(\mathbf{x}_{(\cdot)})}{1 - r^{i_{(\cdot)}}}\right]^{\mathbb{N}_m} \\ &\cdot \sum_{\mathcal{I}_1 \uplus \mathcal{I}_2 = \mathcal{I}_\sigma} \delta_n(|\mathcal{I}_1|) [1_{\mathcal{S}}(\mathbf{x}_{(\cdot)})]^{\{j : i_j \in \mathcal{I}_1\}} [1 - 1_{\mathcal{S}}(\mathbf{x}_{(\cdot)})]^{\{j : i_j \in \mathcal{I}_2\}} \\ &\quad \mathrm{d}\mathbf{x}_1 \cdots \mathrm{d}\mathbf{x}_m, \end{aligned} \tag{109}$$

where the innermost sum is taken over all mutually disjoint subsets $\mathcal{I}_1, \mathcal{I}_2$ such that $\mathcal{I}_1 \cup \mathcal{I}_2 = \mathcal{I}_\sigma$. Distributing terms from the second summation,

$$\begin{aligned}\rho_{\mathcal{S}}(n) &= \left[\left(1 - r^{(\cdot)}\right)\right]^{\mathbb{N}_M} \\ &\cdot \sum_{m=n}^{M} \frac{1}{m!} \int_{\mathbb{X}^m} \sum_{(\mathcal{I}_\sigma) \in \mathbb{K}_M^{(n)}} \delta_m(|\mathcal{I}_\sigma|) \sum_{\mathcal{I}_1 \uplus \mathcal{I}_2 = \mathcal{I}_\sigma} \delta_n(|\mathcal{I}_1|) \\ &\cdot \left[\frac{1_{\mathcal{S}}(\mathbf{x}_{(\cdot)}) r^{i_{(\cdot)}} p^{i_{(\cdot)}}(\mathbf{x}_{(\cdot)})}{1 - r^{i_{(\cdot)}}}\right]^{\{j : i_j \in \mathcal{I}_1\}} \\ &\cdot \left[\frac{[1 - 1_{\mathcal{S}}(\mathbf{x}_{(\cdot)})] r^{i_{(\cdot)}} p^{i_{(\cdot)}}(\mathbf{x}_{(\cdot)})}{1 - r^{i_{(\cdot)}}}\right]^{\{j : i_j \in \mathcal{I}_2\}} \mathrm{d}\mathbf{x}_1 \cdots \mathrm{d}\mathbf{x}_m. \end{aligned} \tag{110}$$

Because $\mathcal{I}_1 \cap \mathcal{I}_2 = \emptyset$, then $\{\mathbf{x}_j : i_j \in \mathcal{I}_1\} \cap \{\mathbf{x}_j : i_j \in \mathcal{I}_2\} = \emptyset$ and the integral on $\mathbb{X}^m$ becomes a product of integrals on $\mathbb{X}$, such that

$$\begin{aligned}\rho_{\mathcal{S}}(n) &= \left[\left(1 - r^{(\cdot)}\right)\right]^{\mathbb{N}_M} \\ &\cdot \sum_{m=n}^{M} \frac{1}{m!} \sum_{(\mathcal{I}_\sigma) \in \mathbb{K}_M^{(n)}} \delta_m(|\mathcal{I}_\sigma|) \sum_{\mathcal{I}_1 \uplus \mathcal{I}_2 = \mathcal{I}_\sigma} \delta_n(|\mathcal{I}_1|) \\ &\cdot \left[\frac{\langle 1_{\mathcal{S}}, r^{i_{(\cdot)}} p^{i_{(\cdot)}} \rangle}{1 - r^{i_{(\cdot)}}}\right]^{\{j : i_j \in \mathcal{I}_1\}} \left[\frac{\langle 1 - 1_{\mathcal{S}}, r^{i_{(\cdot)}} p^{i_{(\cdot)}} \rangle}{1 - r^{i_{(\cdot)}}}\right]^{\{j : i_j \in \mathcal{I}_2\}}. \end{aligned} \tag{111}$$

Now note that the result of the innermost sum does not depend on the permutation order of $(\mathcal{I}_\sigma)$. Thus, the property [51, Lemma 12], which states that for an arbitrary symmetric function h,

$$\sum_{(i_1, \ldots, i_m)} h(\{i_1, \ldots, i_m\}) = m! \sum_{\{i_1, \ldots, i_m\}} h(\{i_1, \ldots, i_m\}) \tag{112}$$

is applied, yielding

$$\begin{aligned}\rho_{\mathcal{S}}(n) &= \left[\left(1 - r^{(\cdot)}\right)\right]^{\mathbb{N}_M} \\ &\cdot \sum_{m=n}^{M} \sum_{\mathcal{I}_1 \uplus \mathcal{I}_2 \subseteq \mathbb{N}_M} \delta_m(|\mathcal{I}_1 \uplus \mathcal{I}_2|) \delta_n(|\mathcal{I}_1|) \\ &\cdot \left[\frac{\langle 1_{\mathcal{S}}, r^{(\cdot)} p^{(\cdot)} \rangle}{1 - r^{(\cdot)}}\right]^{\mathcal{I}_1} \left[\frac{\langle 1 - 1_{\mathcal{S}}, r^{(\cdot)} p^{(\cdot)} \rangle}{1 - r^{(\cdot)}}\right]^{\mathcal{I}_2}. \end{aligned} \tag{113}$$

The term $\delta_m(|\mathcal{I}_1 \uplus \mathcal{I}_2|)$ is nonzero only when the combined cardinality of $\mathcal{I}_1$ and $\mathcal{I}_2$ is equal to m—the index of the outermost sum. Thus, the outermost sum is absorbed by the second sum to give (88). □

REFERENCES

[1] R. P. Mahler
Statistical Multisource-Multitarget Information Fusion. Boston, MA, USA: Artech House, 2007.

[2] B.-N. Vo, B.-T. Vo, and D. Phung
"Labeled random finite sets and the Bayes multi-target tracking filter,"
IEEE Trans. Signal Process., vol. 62, no. 24, pp. 6554–6567, Dec. 2014.

[3] S. Reuter, B.-T. Vo, B.-N. Vo, and K. Dietmayer
"The labeled multi-Bernoulli filter,"
IEEE Trans. Signal Process., vol. 62, no. 12, pp. 3246–3260, Jun. 2014.

[4] Á. F. García-Fernández, Y. Xia, K. Granström, L. Svensson, and J. L. Williams
"Gaussian implementation of the multi-Bernoulli mixture filter,"
in *Proc. 22nd Int. Conf. Inf. Fusion (FUSION)*, 2019, pp. 1–8.

[5] H. G. Hoang and B.-T. Vo
"Sensor management for multi-target tracking via multi-Bernoulli filtering,"
Automatica, vol. 50, no. 4, pp. 1135–1142, 2014. doi: 10.1016/j.automatica.2014.02.007.

[6] M. Beard, B.-T. Vo, B.-N. Vo, and S. Arulampalam
"Void probabilities and Cauchy–Schwarz divergence for generalized labeled multi-bernoulli models,"
IEEE Trans. Signal Process., vol. 65, no. 19, pp. 5047–5061, Oct. 2017.

[7] X. Wang, R. Hoseinnezhad, A. K. Gostar, T. Rathnayake, B. Xu, and A. Bab-Hadiashar
"Multi-sensor control for multiobject Bayes filters," *Signal Process.*, vol. 142, pp. 260–270, Jan. 2018. doi: 10.1016/j.sigpro.2017.07.031.

[8] S. Ferrari, R. Fierro, B. Perteet, C. Cai, and K. Baumgartner
"A geometric optimization approach to detecting and intercepting dynamic targets using a mobile sensor network," *SIAM J. Control Optim.*, vol. 48, no. 1, pp. 292–320, Jan. 2009. doi: 10.1137/07067934X.

[9] H. Wei and S. Ferrari
"A geometric transversals approach to sensor motion planning for tracking maneuvering targets," *IEEE Trans. Autom. Control*, vol. 60, no. 10, pp. 2773–2778, Oct. 2015.

[10] S. Gehly, B. A. Jones, and P. Axelrad
"Search-detect-track sensor allocation for geosynchronous space objects," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 54, no. 6, pp. 2788–2808, Dec. 2018.

[11] A. Buonviri, M. York, K. A. LeGrand, and J. Meub
"Survey of challenges in labeled random finite set based distributed multi-sensor multiobject tracking," in *Proc. IEEE Aerosp. Conf.*, 2019, pp. 1–12.

[12] K. A. LeGrand, P. Zhu, and S. Ferrari
"Cell multi-Bernoulli (cell-MB) sensor control for multiobject search-while-tracking (SWT)," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published, doi: 10.1109/TPAMI.2022.3223856.

[13] K. A. LeGrand, P. Zhu, and S. Ferrari, "A random finite set sensor control approach for vision-based multiobject search-while-tracking," in *Proc. 24th Int. Conf. Inf. Fusion (FUSION)*, 2021, pp. 1–8.

[14] T. Li, V. Elvira, H. Fan, and J. M. Corchado
"Local-diffusion-based distributed SMC-PHD filtering using sensors with limited sensing range," *IEEE Sensors J.*, vol. 19, no. 4, pp. 1580–1589, Feb. 2019.

[15] K. Da, T. Li, Y. Zhu, and Q. Fu
"Gaussian mixture particle Jump-Markov-CPHD fusion for multitarget tracking using sensors with limited views," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 6, pp. 605–616, Aug. 2020.

[16] L. Gao, G. Battistelli, and L. Chisci
"Fusion of labeled RFS densities with different fields of view," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 58, no. 6, pp. 1–16, Dec. 2022.

[17] B. Wang, S. Li, G. Battistelli, L. Chisci, and W. Yi
"Multi-agent fusion with different limited fields-of-view," *IEEE Trans. Signal Process.*, vol. 70, pp. 1560–1575, Mar. 2022.

[18] W. Koch
"On exploiting 'negative' sensor evidence for target tracking and sensor data fusion" *Inf. Fusion*, vol. 8, no. pp. 2839, Jan. 2007.

[19] I. R. Goodman, R. P. Mahler, and H. T. Nguyen
Mathematics of Data Fusion. Norwell, MA, USA: Kluwer Academic Publishers, 1997.

[20] A. Gning, B. Ristic, and L. Mihaylova
"Bernoulli particle/box-particle filters for detection and tracking in the presence of triple measurement uncertainty," *IEEE Trans. Signal Process.*, vol. 60, no. 5, pp. 2138–2151, May 2012.

[21] B. Ristic
Particle Filters for Random Set Models. New York, NY, USA: Springer, Apr. 2013.

[22] A. N. Bishop and B. Ristic
"Fusion of spatially referring natural language statements with random set theoretic likelihoods," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 49, no. 2, pp. 932–944, Apr. 2013.

[23] B. Ristic, B.-T. Vo, B.-N. Vo, and A. Farina
"A tutorial on bernoulli filters: Theory, implementation and applications," *IEEE Trans. Signal Process.*, vol. 61, no. 13, pp. 3406–3430, Jul. 2013.

[24] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp
"A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, Feb. 2002.

[25] B.-N. Vo, S. Singh, and A. Doucet
"Sequential Monte Carlo methods for multitarget filtering with random finite sets," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 41, no. 4, pp. 1224–1245, Oct. 2005.

[26] T. L. Song, D. Musicki, and K. D. Sol
"Target tracking with target state dependent detection," *IEEE Trans. Signal Process.*, vol. 59, no. 3, pp. 1063–1074, Mar. 2011.

[27] N. R. Ahmed, E. M. Sample, and M. Campbell
"Bayesian multicategorical soft data fusion for human-robot collaboration," *IEEE Trans. Robot.*, vol. 29, no. 1, pp. 189–206, Feb. 2013.

[28] B. Wei and B. Nener
"Distributed space debris tracking with consensus labeled random finite set filtering," *Sensors*, vol. 18, pp. 1–26, 2018.

[29] K. A. LeGrand and S. Ferrari
"The role of bounded fields-of-view and negative information in finite set statistics (FISST)," in *Proc. 2020 23rd Int. Conf. Inf. Fusion (FUSION)*, Jul. 2020, pp. 1–9.

[30] K. A. Baumgartner, S. Ferrari, and A. V. Rao
"Optimal control of an underwater sensor network for cooperative target tracking," *IEEE J. Ocean. Eng.*, vol. 34, no. 4, pp. 678–697, Oct. 2009.

[31] B.-N. Vo and W.-K. Ma
"The Gaussian mixture probability hypothesis density filter," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4091–4104, Nov. 2006.

[32] K. A. LeGrand and K. J. DeMars
"Relative multiple space object tracking using intensity filters," in *Proc. 2015 18th Int. Conf. Inf. Fusion*, 2015, pp. 1253–1261.

[33] M. F. Huber, T. Bailey, H. Durrant-Whyte, and U. D. Hanebeck
"On entropy approximation for Gaussian mixture random vectors," *Proc. IEEE Int. Conf. Multisensor Fusion Integration Intell. Syst.*, 2008, pp. 181–188.

[34] K. J. DeMars, R. H. Bishop, and M. K. Jah
"Entropy-based approach for uncertainty propagation of nonlinear dynamical systems," *J. Guidance, Control, Dyn.*, vol. 36, no. 4, pp. 1047–1057, 2013. doi: 10.2514/1.58987.

[35] M. F. Huber
"Adaptive Gaussian mixture filter based on statistical linearization," in *Proc. 14th Int. Conf. Inf. Fusion*, 2011, pp. 1–8.

[36] K. Tuggle and R. Zanetti
"Automated splitting Gaussian mixture nonlinear measurement update," *J. Guid., Control, Dyn.*, vol. 41, no. 3, pp. 1–10, 2018.

[37] R. P. Mahler
Advances in Statistical Multisource-Multitarget Information Fusion. Norwood, MA, USA: Artech House, 2014.

[38] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero, R. L. Moses, and N. S. Correal
"Locating the nodes: Cooperative localization in wireless sensor networks,"
IEEE Signal Process. Mag., vol. 22, no. 4, pp. 54–69, Jul. 2005.

[39] R. Tse and M. Campbell
"Human of probabilistic beliefs via a dirichlet process mixture of statements,"
IEEE Trans. Robot., vol. 34, no. 5, pp. 1280–1298, Oct. 2018.

[40] D. L. Alspach and H. W. Sorenson
"Nonlinear Bayesian estimation using Gaussian sum approximations,"
IEEE Trans. Autom. Control, vol. 17, no. 4, pp. 439–448, Aug. 1972.

[41] M. West
"Approximating posterior distributions by mixtures,"
J. Roy. Stat. Society: Ser. B. (Methodological), vol. 55, no. 2, pp. 409–422, Jan. 1993.

[42] A. R. Runnalls
"Kullback-Leibler approach to Gaussian mixture reduction,"
IEEE Trans. Aerosp. Electron. Syst., vol. 43, no. 3, pp. 989–999, 2007.

[43] D. J. Salmond
"Mixture reduction algorithms for point and extended object tracking in clutter,"
IEEE Trans. Aerosp. Electron. Syst., vol. 45, no. 2, pp. 667–686, Apr. 2009.

[44] D. F. Crouse, P. Willett, K. R. Pattipati, and L. Svensson
"A look at Gaussian mixture reduction algorithms,"
in *Proc. 2011 14th Int. Conf. Inf. Fusion*, 2011, pp. 1–8.

[45] E. Delande, M. Üney, J. Houssineau, and D. E. Clark
"Regional variance for multiobject filtering,"
IEEE Trans. Signal Process., vol. 62, no. 13, pp. 3415–3428, Jul. 2014.

[46] E. Delande, J. Houssineau, and D. Clark
"Regional variance in target number: Analysis and application for multi-Bernoulli point processes,"
in *Proc. IET Conf. Data Fusion Target Tracking* Algorithms Appl., 2014, pp. 4.2–4.2.

[47] K. F. Riley, M. P. Hobson, and S. J. Bence
Mathematical Methods for Physics and Engineering: A. Comprehensive Guide. Cambridge, U.K.: Cambridge Press, 2006.

[48] J. L. Williams
"Marginal multi-Bernoulli filters: RFS derivation of MHT, JIPDA, and association-based MeMBer,"
IEEE Trans. Aerosp. Electron. Syst., vol. 51, no. 3, pp. 1664–1687, Jul. 2015.

[49] M. Fernandez and S. Williams
"Closed-form expression for the Poisson-binomial probability density function,"
IEEE Trans. Aerosp. Electron. Syst., vol. 46, no. 2, pp. 803–817, Apr. 2010.

[50] R. P. Mahler
"PHD filters of higher order in target number,"
IEEE Trans. Aerosp. Electron. Syst., vol. 43, no. 4, pp. 1523–1543, Oct. 2007.

[51] B.-T. Vo and B.-N. Vo
"Labeled random finite sets and multiobject conjugate priors,"
IEEE Trans. Signal Process., vol. 61, no. 13, pp. 3460–3475, Jul. 2013.

Keith LeGrand is an Assistant Professor of Aeronautics and Astronautics at Purdue University, West Lafayette, IN USA. He received the Ph.D. degree in aerospace engineering from Cornell University, Ithaca, NY, USA. Prior to that, he was a Senior Member of Technical Staff at Sandia National Laboratories in Albuquerque, New Mexico, where he conducted research in inertial navigation, space systems, and multiobject tracking. He received the B.S. and M.S. degrees in aerospace engineering from the Missouri University of Science and Technology. He is the recipient of the U.S. DoD National Defense Science and Engineering Graduate (NDSEG) Fellowship (2021) and multiple best paper awards at the International FUSION Conference. His research interests include multisensor, multiobject tracking, space domain awareness, and information theory.

Silva Ferrari is John Brancaccio Professor of Mechanical and Aerospace Engineering at Cornell University, Ithaca, NY, USA. Prior to that, she was a Professor of Engineering and Computer Science at Duke University, Durham, NC, USA, and Founder and Director of the NSF Integrative Graduate Education and Research Traineeship (IGERT) and Fellowship program on Wireless Intelligent Sensor Networks (WISeNet). She is currently the Director of the Laboratory for Intelligent Systems and Controls (LISC) at Cornell University, Ithaca, NY, USA, and the Co-Director of the Cornell-Unibo Věho Institute on Vehicle Intelligence at Cornell Tech, New York, NY, USA. Her principal research interests include active perception, robust adaptive control, learning, and approximate dynamic programming, and control of multiscale dynamical systems. She is the author of the book *Information-Driven Path Planning and Control (MIT Press, 2021)*, and of the TED talk "Do robots dreams of electric sheep?" She received the B.S. degree from Embry-Riddle Aeronautical University and the M.A. and Ph.D. degrees from Princeton University. She is a Fellow of ASME, a Senior Member of the IEEE and AIAA, and a Member of SPIE and SIAM. She is the recipient of the ONR young investigator award (2004), the NSF CAREER award (2005), and the Presidential Early Career Award for Scientists and Engineers (PECASE) award (2006).

Variations of Joint Integrated Data Association With Radar and Target-Provided Measurements

AUDUN G. HEM
EDMUND F. BREKKE

Target tracking algorithms are usually based on exteroceptive measurements obtained from sensors placed in the center of some surveillance area. However, information transmitted from surrounding targets will often also be available. This information, here dubbed target-provided measurements, will often include valuable information for a tracking system. We present a multitarget tracking algorithm utilizing such measurements using a framework of joint integrated data association. The use case we consider is maritime target tracking using radar measurements combined with messages from the automatic identification system. The full details of the tracking algorithm are presented, including implementation-specific considerations to account for the different natures of the incoming measurements. We detail three different methods of handling the target-provided measurements: one processing them as they arrive, i.e., sequentially, and the others collecting and processing them at fixed intervals. The results show that all three improve over the pure radar tracking algorithm and similar state-of-the-art methods.

Manuscript received January 19, 2022; revised January 26, 2023; released for publication March 20, 2023.

Audun G. Hem and Edmund F. Brekke are with the Department of Technical Cybernetics, Norwegian University of Science and Technology, 7034 Trondheim, Norway (e-mail: audun.g.hem@ntnu.no; edmund.brekke@ntnu.no).

Refereeing of this contribution was handled by Ting Yuan.

This work was supported in part by the Research Council of Norway through Projects 223254 AMOS, 309230 SFI Autoship and 295033 Autosit, in part by DNV, in part by Kongsberg, and in part by Maritime Robotics.

1557-6418/22/$17.00 © 2022 JAIF

I. INTRODUCTION

One of the many important puzzle pieces for increased degrees of autonomy in the maritime sector is the ability of a ship to observe its surroundings. To avoid collisions and safely navigate the waters, it is necessary to know where the surrounding ships are situated. For this to work safely and robustly, target tracking algorithms have to provide precise estimates of the position and direction of surrounding vessels, also known as targets. Radar-based target tracking algorithms have largely been the norm when navigating outside of close encounter harbor areas. There is, however, also a standardized system to help with collision avoidance at sea: the automatic identification system (AIS). This system provides target-provided measurements with valuable information that could help give better estimates than what only radar measurement can provide. However, this valuable source of information often remains unused in modern target tracking algorithms.

When monitoring aircraft, target-provided measurements are also used, with measurements based on the automatic dependent surveillance–broadcast (ADS-B) protocol. The latter protocol can, together with radar, be used in air traffic control to provide a better picture of the airspace [4]. The availability of target-provided measurements makes it possible to identify targets and utilize information that is impossible to get from radar measurements alone, such as the ship destination. For, e.g., long-time vessel prediction, the additional information provided by target-provided measurements can be very valuable [31].

The two measurement types are inherently different. The radar is attached to the ship, scanning the surrounding area. The measurements are unlabeled, can be false alarms, and can provide several detections for each target. The last issue is often solved using a clustering algorithm, while the problem of false alarms has no single simple solution. The radar measurements are also often noisier than the target-provided measurements, with the noise becoming more prominent when the target is far away. Target-provided measurements, on the other hand, are sent out from the surrounding ships as data packages containing not only the position of the target but additional information as well, such as the ID number of the transmitting ship. Because a target needs to send a target-provided measurement for it to be received, there are no false alarms, and the precision of the transmitted kinematic information is independent of the distance to the target because the positional data comes from GPS measurements. However, not all targets have a transmitter, and the messages will often be received somewhat infrequently, as high-frequency transmitting is not always required, see, e.g., [19]. Thus, a robust target tracking system based only on target-provided measurements will not be feasible.

There are two established approaches to the fusion of sensor signals: track-to-track fusion and track-to-measurement fusion [1]. Here track-to-measurement fusion is examined, and a model suitable for incorporating target-provided measurements, and a tracking algorithm utilizing this model, is presented. For example, Gaglione *et al.* [13] have previously investigated track-to-measurement fusion for radar and target-provided information. The tracking algorithm presented here differs from previous work in some significant ways. We use a hybrid state framework based on [7], which can include motion and visibility models in addition to target IDs. Furthermore, building upon [7], we derive the tracking algorithm as a special case of the Poisson multi-Bernoulli mixture (PMBM) filter originally proposed in [34]. An important technical detail to enable this is to model the birth model as a marked Poisson point process (PPP), where the target IDs take the role of the marks. The resulting algorithm can be seen as a generalized version of joint integrated probabilistic data association (JIPDA) [24].

The contributions of this paper are as follows. It derives a framework that includes target-provided measurements based on a PMBM formulation of the JIPDA. The resulting target tracker includes both a visibility state and multiple kinematic models. Furthermore, the paper details a sequential way of handling the incoming target-provided measurements, a method more similar to the one described in [13], and a method similar to how radar measurements are processed. Lastly, we present some implementation-specific considerations to make when handling target-provided measurements in a tracker.

The paper is organized as follows: We detail the problem formulation in Section III. In Section IV, we explain the structure of the hybrid state that facilitates the inclusion of target-provided information. We present the mathematical expressions needed for calculations in Section V. In Section VI, three different methods for handling the incoming measurements are detailed. Section VII presents the implementation choices, together with considerations to make to accommodate the target-provided measurements. Lastly, Section VIII presents the results. We compare the performance of the different measurement handling methods and how they compare to using only radar and the method from [13].

II. BACKGROUND

This work builds upon the multitarget tracking method presented in [7] and can be considered an extension of the framework described there. The tracking algorithm, denoted as visibility interacting multiple models joint integrated probabilistic data association (VIMMJIPDA), combines interacting multiple models (IMM) and a visibility state with the well-established JIPDA framework. The tracking method was derived with a basis in the PMBM filter [34].

Darko Musicki and Rob Evans introduced the JIPDA in [24], where the concept of visibility is mentioned and indicates whether the tracked target is visible to the sensor or not. Later, e.g., [35] has explored visibility in connection with the problem of estimating target detectability. The JIPDA is an extension of the joint probabilistic data association (JPDA) method developed by Yaakov Bar-Shalom [12], which again is an extension of Bar-Shalom's probabilistic data association (PDA) method [3]. These methods are well established in the target tracking community and have been used for a range of different purposes, such as collision avoidance for marine vessels [29], autonomous navigation [11], and air traffic control [20]. Henk A. P. Blom and Yaakov Bar-Shalom introduced the IMM method [5], and it has been used for several decades in, e.g., air traffic control. Furthermore, Musicki and Suvorova presented an IMM-JIPDA algorithm in [25].

The PMBM filter and subsequent tracking algorithms [15] utilize the PMBM density, which is the union between a PPP and a multi-Bernoulli mixture (MBM). The PPP represents unknown targets, i.e., undetected targets hypothesized to exist, and the MBM represents already detected targets. Links between PMBM and JIPDA have been established in [34] (single kinematic model, loopy belief propagation as an alternative to hypothesis enumeration) and in [7] (multiple kinematic models, standard hypothesis enumeration, and mixture reduction).

Some work on the track-to-measurement fusion of radar and target-provided measurements has been done previously, both by Habtemariam *et al.* [17] and Gaglione *et al.* [13]. The first approach includes target-provided measurements in a JPDA-like tracking algorithm, while the second uses a framework that also includes track existence. The second approach utilizes probabilistic graphical models and loopy belief propagation for the calculations. Furthermore, Gaglione *et al.* use particle filtering for performing the calculations. Both works perform data association on batches of target-provided measurements simultaneously as on the radar measurements. Gaglione *et al.* nevertheless consider that target-provided measurements can arrive at any time. They also share similar modeling of the target-provided measurement IDs, from which the model presented here deviates. However, neither method directly addresses the initialization of tracks using target-provided measurements. In [21], a multiple hypothesis tracking (MHT) approach is presented, which also showed promising results but relied on preprocessing of the AIS measurements. Track-to-track fusion using radar and AIS measurements has also been done previously, e.g., in [9]. Here, a multisensor network for maritime surveillance is described, utilizing several sensors, including radar and AIS. More recently, research has been conducted into the track-to-track association of radar- and AIS-tracks [27].

III. PROBLEM FORMULATION

The unknown target intensity $u(\mathbf{y})$ describes the not yet discovered targets present in the surveillance area. We model the unknown targets as a marked PPP, which is equivalent to a PPP on the Cartesian product of the space $\mathbf{R}^{n_x}$ and the discrete spaces the discrete hybrid states can take values from [30, p. 205]. In its general form, this process is

$$b(\mathbf{y}) = p(v)p(\tau|v)p(s|v,\tau)p_{\gamma}(\mathbf{x}|s,v,\tau), \tag{1}$$

where $p_{\gamma}(\mathbf{x}|s,v,\tau)$ is an intensity function on the the space $\mathbf{R}^{n_x}$, and $p(\cdot)$ are distributions over the discrete states. Rather than using the birth intensity directly, we use Proposition 1 from [7] to get the converged unknown target intensity

$$u(\mathbf{y}) = U o_u^v \xi_u^\tau \mu_u^\tau f_u(\mathbf{x}). \tag{2}$$

Here, U is the overall birth rate of new targets, o_u^v is the probability of visibility state v, ξ_u^τ is the probability of ID τ, μ_u^τ is the probability of the kinematic mode s, and $f_u(\mathbf{x})$ is the distribution of the kinematic state. The subscript u indicates that the individual expressions are part of the unknown target intensity. Equation (2) does not contain the initial values of new targets, as it is a function of the birth intensity and the transition probability matrices. However, for simplicity, the unknown target values are tuned directly and can be viewed as initial values.

Remark 1. This method of modeling the target IDs through a marked PPP implies that two targets can have the same ID. The probability of two targets having the same ID in a surveillance area with relatively few targets is minuscule, but it is nevertheless a possibility [10]. We also note how the modeling of actual, observable IDs here deviates from theoretically assigned IDs. The labels in labeled random finite sets (RFSs), introduced in [32], are unobservable and analogous to the identifying tags in [14], which ensure the uniqueness of the elements of a RFS. The IDs described here, however, serve no such purpose and can be assumed nonunique without breaking the underlying mathematical assumptions of RFSs.

M2: We model the survival probability as a function of time since the last update. A constant parameter P_{S_c} denotes the probability of survival after one second. Thus, the survival probability of an interval between times t_{k-1} and t_k, denoted as Δt, becomes

$$P_S(\Delta t) = P_{S_c}^{\Delta t}. \tag{3}$$

M3: The ID numbers τ are assumed to be static, in line with the physical reality of the AIS protocol. The IDs are manually set at the installation of the AIS system. We assume that the ID numbers of the unknown targets are distributed according to

$$\xi_u^\tau = \begin{cases} \xi_u^0 & \text{if } \tau = 0 \\ \dfrac{1-\xi_u^0}{|\mathcal{V}|-1} & \text{if } \tau > 0 \end{cases}, \tag{4}$$

where ξ_u^0 is some parameter denoting the belief that the target has no ID and $|\mathcal{V}|$ is the number of all possible ID numbers in addition to 0. Not all targets have an ID, and we represent this non-ID by the value $\tau = 0$. If $\tau = 0$, the target does not transmit measurements.

M4: From time step $k-1$ to k, the evolution of a target is given by

$$f_{\mathbf{y}}(\mathbf{y}_k|\mathbf{y}_{k-1}) = f_{\mathbf{x}}^{s\tau}(\mathbf{x}_k|\mathbf{x}_{k-1})\pi^{s_{k-1}s_k} w^{v_{k-1}v_k}. \tag{5}$$

The π-matrix contains the Markov chain probabilities of changing between different kinematic models. The matrix w contains the Markov chain probabilities of the target switching between the visible state $v = 1$ and invisible state $v = 0$. The ID numbers are assumed static and therefore do not change during a prediction.

M5: For radar measurements, the detection probability $P_D(\mathbf{y}_k)$ varies based on the visibility state v, and we define it as

$$P_D(\mathbf{y}_k) = \begin{cases} P_D & \text{if } v = 1 \\ 0 & \text{if } v = 0 \end{cases}, \tag{6}$$

where P_D is a constant describing the probability of a target being detected by the radar at a given time step.

For target-provided measurements, which are assumed to give no missed detections, we have that

$$P_D(\mathbf{y}_k) = \begin{cases} 1 & \text{if a target-provided measurement} \\ & \quad \text{is received} \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

independent of the visibility state. Thus, no conclusions about a target are made from the absence of target-provided measurements. Trying to keep track of when a vessel should transmit measurements is a difficult problem that, e.g., would be subject to intentional randomness from the protocol [6].

M6: Radar clutter measurements are assumed to follow a Poisson process with intensity λ. The target-provided measurements do not contain clutter, the same as if they are following a Poisson process with intensity 0.

M7: The radar measurements are assumed to be synchronized and to arrive simultaneously at a fixed frequency. The synchronicity means that when radar measurements arrive at time step k, the set of radar measurements contains measurements from all detected targets at time step k, in addition to clutter measurements. The radar measurement likelihood is denoted as $f_{\mathbf{z}}^R(\mathbf{z}_k|\mathbf{y}_k)$.

M8: The target-provided measurements can arrive whenever and are not synchronized. Thus, a transmitted measurement can be received at any time from any target. We do not assume that targets transmit measurements simultaneously, contrary to what we do for radar measurements. Whenever a target-provided measurement arrives, however, the time of arrival is assumed to be known. The measurement likelihood for the target-provided measurements is

$$f_{\mathbf{z}}^{A}(\mathbf{z}_k|\mathbf{y}_k) = f_{\mathbf{p}}(\mathbf{p}_k|\mathbf{y}_k) f_\tau(\tau^{\mathbf{z}_k}|\tau), \tag{8}$$

where $\mathbf{z}_k$ is the whole measurement and $\mathbf{p}_k$ only contains the kinematic data of the measurement. Furthermore,

$$f_\tau(\tau^{\mathbf{z}_k}|\tau) = \begin{cases} P_C & \text{if } \tau_k = \tau_k^{\mathbf{z}_k} \\ \dfrac{1-P_C}{|\mathcal{V}|-1} & \text{if } \tau_k \neq \tau_k^{\mathbf{z}_k} \text{ and } \tau > 0, \\ 0 & \text{if } \tau = 0 \end{cases} \tag{9}$$

where P_C is a fixed parameter describing the confidence in the ID number not being corrupted, denoted as the confidence probability. The reasoning behind the above equation comes from the observation that the likelihood of a transmitted measurement coming from a target without an ID is zero. Furthermore, the chance of a transmitted ID being erroneous makes it a possibility, albeit small, that any ID can be the correct one.

IV. HYBRID STATES AND THE PMBM

As formulated in [2, p. 441], a hybrid state is a state where the state space contains both discrete and continuous states or uncertainties. This structure is useful as the kinematic state will be continuous, while, e.g., the choice of kinematic model for the target will be discrete.

A PMBM filter represents the posterior multitarget density for discovered targets as a weighted sum of multi-Bernoulli densities. These involve weights for each of the multi-Bernoullis, and kinematic densities and existence probabilities for each of the Bernoullis. The PMB filter, which is essentially the same as a JIPDA, approximates the sum of multi-Bernoullis by a single multi-Bernoulli at the end of each estimation cycle.

Using the equations from [34], one can get general expressions for the weight, existence, and states irrespective of the sensor type, assuming the sensors generate measurements adhering to the assumptions made in Assumption 2 in [34]. The assumptions hold for both target-provided and radar measurements. The inclusion of IDs in the target-provided measurements is contained in the measurement likelihood function, and they do not breach any independence assumptions. The goal of this section is to extract expressions for the probabilistic properties of the individual hybrid state elements.

From [34], we have that the weight w, existence probability r, and distribution $f(\mathbf{y})$ of a single Bernoulli in general can be written as

$$w = g(\mathbf{y}) + h[1], \tag{10}$$

$$r = \frac{h[1]}{g(\mathbf{y}) + h[1]}, \tag{11}$$

$$f(\mathbf{y}) = \frac{h(\mathbf{y})}{h[1]} \tag{12}$$

for some functions g and h of the state $\mathbf{y}$. The notation $[\cdot]$ indicates a linear functional, defined as

$$g[h] = \int g(\mathbf{x}) h(\mathbf{x}) d\mathbf{x}. \tag{13}$$

These are useful tools for compactly writing normalization constants and likelihoods. For later use, it is convenient to find general expressions for the individual states in the hybrid state $\mathbf{y}$. Using the approximation from [7, Remark 6] that the visibility is independent on the other states, we can write $h(\mathbf{y}) = h(v)h(\tau)h(s|\tau)h(\mathbf{x}|\tau, s)$. We get the individual states by using the rule of conditional probability. Starting with the kinematic state $\mathbf{x}$, it can be acquired by

$$\begin{aligned} f^t(\mathbf{x}|s, \tau, v) &= \frac{f(\mathbf{x}, s, \tau, v)}{\int f(\tilde{\mathbf{x}}, s, \tau, v) \mathrm{d}\tilde{\mathbf{x}}} \\ &= \frac{\dfrac{h(\mathbf{x}, s, \tau, v)}{h[1]}}{\dfrac{\int h(\tilde{\mathbf{x}}, s, \tau, v) \mathrm{d}\tilde{\mathbf{x}}}{h[1]}} \\ &= \frac{h(\mathbf{x}, s, \tau, v)}{\int h(\mathbf{x}, s, \tau, v) \mathrm{d}\mathbf{x}} \\ &= \frac{h(v) h(\mathbf{x}, s, \tau)}{h(v) \int h(\tilde{\mathbf{x}}, s, \tau, v) \mathrm{d}\tilde{\mathbf{x}}} \\ &= \frac{h(\mathbf{x}, s, \tau)}{h(s, \tau)}, \end{aligned} \tag{14}$$

where we have omitted the time indices for brevity. The $\tilde{(\cdot)}$ notation is used for latent variables, which disappear by marginalization. Furthermore, the absence of the visibility state v in the final expression means that $f^t(\mathbf{x}|s, \tau, v) = f^t(\mathbf{x}|s, \tau)$. Similarly, the mode probabilities are

$$f^t(s|\tau) = \mu^{t\tau s} = \frac{h(s, \tau)}{h(\tau)}, \tag{15}$$

the ID probabilities are

$$f^t(\tau) = \xi^{t\tau} = \frac{h(\tau)}{h[1]}, \tag{16}$$

and the visibility probabilities are

$$f^t(v) = o^{tv} = \frac{h(v)}{h[1]}. \tag{17}$$

Note that $\sum_{\tilde{\tau}} \sum_{\tilde{s}} \int h(\tilde{\mathbf{x}}, \tilde{s}, \tilde{\tau}) \mathrm{d}\tilde{\mathbf{x}} = h[1]$, which essentially acts as a normalization constant. Independencies between the states will make it possible to reduce the needed amount of marginalization, as they will appear both in the numerator and the denominator. The independencies will depend on the model choices and are written here according to the assumptions in Section III.

V. INCLUDING TARGET-PROVIDED MEASUREMENTS IN THE VIMMJIPDA

In the VIMMJIPDA, the unknown target intensity $u(\mathbf{y})$ is assumed stationary and is left unchanged during the prediction and updating of the estimates. We make the same assumption here. This assumption means that only the Bernoulli components have to be considered, and is further simplified by following the JIPDA method of performing mixture reduction. That is, we merge all Bernoullis originating in the same measurement into a single Bernoulli after each update. Thus, we can omit the weights of the association hypotheses of previous time steps can due to marginalization. Table II shows the expressions for updating and predicting the Bernoulli components from [34]. These are adapted to simplify insertion in (10)–(12) and (14)–(17). Furthermore, they are simplified to reflect the stationary unknown target intensity and the marginalization over the weights during mixture reduction. As the measurement model assumptions made in [34] hold with regards to both radar and target-provided measurements, both $f^R_{\mathbf{z}}(\mathbf{z}|\mathbf{y})$ and $f^A_{\mathbf{z}}(\mathbf{z}|\mathbf{y})$ can be considered special cases of the more general $f_{\mathbf{z}}(\mathbf{z}|\mathbf{y})$ in the table. The expressions for predicting and updating the Bernoulli estimates based on the potential information acquired by the sensor updates follow.

A. Prior

For a single track, which in the context of this paper is analogous to a Bernoulli, we write the hybrid state prior distribution as

$$f^t_{k-1}(\mathbf{y}) = f^t_{k-1}(\mathbf{x}|\tau, s)\xi^{t\tau}_{k-1}\mu^{t\tau s}_{k-1}o^{tv}_{k-1}, \tag{18}$$

while the prior existence probability is r^t_{k-1}. As mentioned above, we merge all the hypotheses of the previous time step, giving $w^t_{k-1} = 1$. The prior is a joint distribution over the continuous kinematic state and the discrete potential IDs, kinematic modes, and visibility states. In the following propositions, only the probability of the target being in the visible state is presented, i.e., o^{t1}, which we denote as η^t. The prior is decomposed into several states conditioned on the different discrete states. An example of the structure of a prior with two possible IDs and two possible kinematic modes is shown in Fig. 1. The expressions in the square boxes are not calculated themselves but can be constructed from the other expressions.

B. Prediction

All tracks are predicted from the previous time step $k-1$ to the current time step k. The predicted probabilities and densities are denoted by the subscript $(\cdot)_{k|k-1}$.

Proposition 1. *The prediction for the existence probability r^t, the visibility probability η^t, the ID probabilities $\xi^{t\tau}$, the mode probabilities $\mu^{t\tau s}$, and the kinematic density $f^t(\mathbf{x}|\tau, s)$ are done as*

$$r^t_{k|k-1} = r^t_{k-1}P_S(\Delta t), \tag{19}$$

$$\eta^t_{k|k-1} = (1-\eta^t_{k-1})w^{01} + \eta^t_{k-1}w^{11}, \tag{20}$$

$$\xi^{t\tau}_{k|k-1} = \xi^{t\tau}_{k-1}, \tag{21}$$

$$\mu^{t\tau s}_{k|k-1} = \sum_{\tilde{s}} \mu^{t\tau\tilde{s}}_{k-1}\pi^{\tilde{s}s}(\Delta t), \tag{22}$$

Table I
Nomenclature

$\mathbf{a}$	Association hypothesis
$b(\cdot)$	Birth intensity function
$1_\Omega(\cdot)$	Indicator function
$\mathbf{H}$	Measurement matrix
$\mathbf{H}^*$	Complementary measurement matrix
$\mathcal{N}(\cdot)$	Gaussian probability density function
μ	Mode probabilities
η	Probability of a target being visible
ξ	ID probabilities
$f(\cdot)$	Generic (single-target) probability density function (pdf)
$f_{\mathbf{y}}(\cdot)$	Transition density for hybrid state
$f_{\mathbf{z}}(\cdot)$	Measurement density conditional on hybrid state
$\mathbf{F}$	Process model transition matrix
$g[h]$	functional with test function
h	Generic hybrid state probability density function
j	Measurement index (superscript)
k	Time step index (subscript)
$\Delta\mathrm{t}$	Interval between current and preceding time step
λ	Poisson intensity for false alarms
n	Number of tracks
o	Visibility probabilities
P_{S_c}	Constant survival probability
P_{D}	Detection probability
$\mathbf{P}_v$	Initial velocity covariance
π	Mode transition probabilities
$\mathbf{Q}$	Process noise covariance matrix
r	Existence probability
$\mathbf{R}$	Measurement noise covariance matrix
$\mathbf{R}_c$	Cartesian measurement noise covariance contribution
$\mathbf{R}_p$	Polar measurement noise covariance contribution
s	Model index (superscript)
τ	ID number (superscript)
t	Track index (superscript)
u	Poisson intensity of unknown targets
U	Unknown target intensity strength
v	Visibility state (superscript)
$\mathbf{v}$	Process noise
$\mathbf{w}$	Measurement noise
ω	Visibility transition probabilities *or* turn rate
Ω	Surveillance region
x	Kinematic (continuous) state vector
$\mathbf{y}$	Hybrid state vector
$\mathbf{z}$	Measurement vector
A	Target-provided (AIS) specific entity
R	Radar specific entity
$(\cdot)_k$	A (typically posterior) quantity at time step k
$(\cdot)_{k\|k-1}$	A predicted quantity at time step k
$(\hat{\cdot})$	A Kalman filter estimate
$(\tilde{\cdot})$	Latent variables that are marginalized away
$(\cdot)^0$	An initial quantity. Further meaning is context-dependent.
$(\cdot)_u$	Unknown target intensity parameter after convergence

Table II
Expressions for Creating, Updating, and Predicting the Bernoulli Components

	g	$h[1]$	$h(\mathbf{y})$
New target	λ	$u[P_D(\tilde{\mathbf{y}})f_{\mathbf{z}}(\mathbf{z}\|\tilde{\mathbf{y}})]$	$u(\mathbf{y})P_D(\mathbf{y})f_{\mathbf{z}}(\mathbf{z}\|\mathbf{y})$
Missed detection	$1 - r^t_{k\|k-1}$	$r^t_{k\|k-1}f[1 - P_D(\tilde{\mathbf{y}})]$	$r^t_{k\|k-1}f_{k\|k-1}(\mathbf{y})(1 - P_D(\mathbf{y}))$
Detection	0	$r^t_{k\|k-1}f[P_D(\tilde{\mathbf{y}})f_{\mathbf{z}}(\mathbf{z}\|\tilde{\mathbf{y}})]$	$r^t_{k\|k-1}f^t_{k\|k-1}(\mathbf{y})P_D(\mathbf{y})f_{\mathbf{z}}(\mathbf{z}\|\mathbf{y})$
Prediction	$1 - r^t_{k-1}f[P_S(\tilde{\mathbf{y}})]$	$r^t_{k-1}f[P_S(\tilde{\mathbf{y}})]$	$r^t_{k-1}\int f^t_{k\|k-1}(\mathbf{y}\|\tilde{\mathbf{y}})P_S(\tilde{\mathbf{y}})f_{k-1}(\tilde{\mathbf{y}}))\mathrm{d}\tilde{\mathbf{y}}$

$$f^t_{k|k-1}(\mathbf{x}|\tau, s) = \int f_{\mathbf{y}}(\mathbf{x}|\tau, s, \tilde{\mathbf{x}})f^t_{k-1}(\tilde{\mathbf{x}}|\tau, s)\mathrm{d}\tilde{\mathbf{x}}, \tag{23}$$

where

$$f^t_{k-1}(\tilde{\mathbf{x}}|\tau, s) = \sum_{\tilde{s}} \frac{\mu^{t\tau\tilde{s}}_{k-1}\pi^{\tilde{s}s}f^t_{k-1}(\tilde{\mathbf{x}}|\tau, \tilde{s})}{\sum_{\tilde{s}}\mu^{t\tau\tilde{s}}_{k-1}\pi^{\tilde{s}s}(\Delta t)}. \tag{24}$$

Proof. The proof builds upon [7], but is modified to also account for the inclusion of the IDs in the state vector. It should be noted that the survival probability is only dependent on the times of the measurements' arrival, which are independent of the state. Because the IDs are assumed to be static the transition model for the IDs becomes a Kronecker delta $\delta_{\tau\tilde{\tau}}$. It is defined as

$$\delta_{\tau\tilde{\tau}} = \begin{cases} 1 & \text{if } \tau = \tilde{\tau} \\ 0 & \text{if } \tau \neq \tilde{\tau} \end{cases}. \tag{25}$$

First, we write out $h(\mathbf{y})$ from Table II:

$$\begin{aligned} h(\mathbf{y}) &= r^t_{k-1}\int f^t_{k|k-1}(\mathbf{y}|\tilde{\mathbf{y}})P_S(\tilde{\mathbf{y}})f_{k-1}(\tilde{\mathbf{y}})\mathrm{d}\tilde{\mathbf{y}} \\ &= r^t_{k-1}P_S(\Delta t)\Big(\sum_{\tilde{v}} f(\tilde{v})f(v|\tilde{v})\Big)\times \\ &\quad \times \sum_{\tilde{\tau}} f_{k-1}(\tilde{\tau})\delta_{\tau\tilde{\tau}}\sum_{\tilde{s}} f_{k-1}(\tilde{s}|\tilde{\tau})f^t_{k|k-1}(s|\tilde{s})\times \\ &\quad \times \int f^t_{k|k-1}(\mathbf{x}|s, \tau, \tilde{\mathbf{x}})f_{k-1}(\tilde{\mathbf{x}}|\tilde{s}, \tilde{\tau})\mathrm{d}\tilde{\mathbf{x}} \\ &= r^t_{k-1}P_S(\Delta t)\Big(\sum_{\tilde{v}} f(\tilde{v})f(v|\tilde{v})\Big)f_{k-1}(\tau)\times \\ &\quad \times \sum_{\tilde{s}} f_{k-1}(\tilde{s}|\tau)f^t_{k|k-1}(s|\tilde{s})\int f^t_{k|k-1}(\mathbf{x}|s, \tau, \tilde{\mathbf{x}})\times \\ &\quad \times f_{k-1}(\tilde{\mathbf{x}}|\tilde{s}, \tau)\mathrm{d}\tilde{\mathbf{x}} \\ &= r^t_{k-1}P_S(\Delta t)\Big(\sum_{\tilde{v}} o^{t\tilde{v}}_{k-1}w^{\tilde{v}v}\Big)\xi^{t\tau}_{k-1}\sum_{\tilde{s}}\mu^{\tau s}_{k-1}\pi^{\tilde{s}s}(\Delta t)\times \\ &\quad \times \int f^{t\tau s}_{k|k-1}(\mathbf{x}|s, \tau, \tilde{\mathbf{x}})f_{k-1}(\tilde{\mathbf{x}}|\tilde{s}, \tau)\mathrm{d}\tilde{\mathbf{x}}, \end{aligned} \tag{26}$$

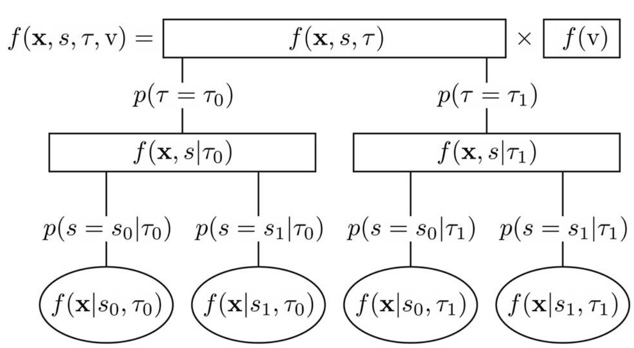


Fig. 1. The structure of the distribution of a hybrid state with two kinematic modes and two possible IDs.

which uses the fact that only the conditioning on the most recent variable is relevant. Marginalizing this, one gets

$$\begin{aligned} h(s, \tau) &= r^t_{k-1}\int\sum_v h(\mathbf{x}, s, \tau, v)\mathrm{d}\mathbf{x} \\ &= r^t_{k-1}P_S(\Delta t)\xi^{t\tau}_{k-1}\sum_{\tilde{s}}\mu^{t\tau\tilde{s}}_{k-1}\pi^{\tilde{s}s}(\Delta t), \end{aligned} \tag{27}$$

$$h(\tau) = r^t_{k-1}\sum_s h(s, \tau) = P_S(\Delta t)\xi^{t\tau}_{k-1}, \tag{28}$$

$$\begin{aligned} h(v) &= r^t_{k-1}\int\sum_\tau\sum_s h(\mathbf{x}, s, \tau, v)\mathrm{d}\mathbf{x} \\ &= r^t_{k-1}P_S(\Delta t)\Big(\sum_{\tilde{v}} o^{t\tilde{v}}_{k-1}w^{\tilde{v}v}\Big) \end{aligned} \tag{29}$$

$$h[1] = r^t_{k-1}\sum_\tau h(\tau) = r^t_{k-1}P_S(\Delta t). \tag{30}$$

Inserting this in (14)–(17) provides the expressions for the hybrid states. Note that the expression for the visibility probability $\eta^t_{k|k-1}$ follows from the fact that $o^{t0}_{k-1} = 1 - o^{t1}_{k-1} = 1 - \eta^t_{k-1}$. The expression for the existence probability $r^t_{k|k-1}$ is found by inserting $g(\mathbf{y}) = r^t_{k-1}P_S(\Delta t)$ from Table II and $h[1]$ into (11). □

C. Posterior

The individual posterior distributions, conditioned on either a detection or a missed detection, are calculated after the prediction. The four possibilities for a track when new measurements arrive are

- The previously unknown track is detected for the first time.
- The previously detected track is detected again.
- The previously detected track is not detected.
- The previously unknown track is not detected.

Any tracks covered by the fourth alternative will be represented by the unknown target density, and do not need to be considered specifically. The posterior distributions for the three first possibilities are presented in the following propositions.

Proposition 2. *Initialization of a new track on a measurement indexed by j is done as*

$$w_k^{tj} = \begin{cases} \lambda + cUP_D\eta^0 & \text{for radar} \\ cU\sum_{\tilde{\tau}}\xi_u^{\tilde{\tau}} f_\tau(\tau^j|\tilde{\tau}) & \text{for target-provided} \end{cases}, \quad (31)$$

$$r_k^{tj} = \begin{cases} \dfrac{UP_D\eta^0}{\lambda + UP_D\eta^0} & \text{for radar} \\ 1 & \text{for target-provided} \end{cases}, \quad (32)$$

$$\eta_k^{tj} = \begin{cases} 1 & \text{for radar} \\ \eta_u & \text{for target-provided} \end{cases}, \quad (33)$$

$$\xi_k^{t\tau j} = \begin{cases} \xi_u^{\tau} & \text{for radar} \\ f_\tau(\tau^{\mathbf{z}}|\tau) & \text{for target-provided} \end{cases}, \quad (34)$$

$$\mu_k^{t\tau s j} = \mu_u^s, \quad (35)$$

$$f_k^{tj}(\mathbf{x}|s,\tau) = f_{\mathbf{z}}(\mathbf{z}|\mathbf{x},s,\tau)f_u(\mathbf{x})/c, \quad (36)$$

where $c = \int f_{\mathbf{z}}(\mathbf{z}|\mathbf{x},s,\tau)f_u(\mathbf{x})\mathrm{d}\mathbf{x}$ is a constant.

Proof. Firstly, for radar measurements, we have that

$$h(\mathbf{y}) = UP_D(v)o_u^v\xi_u^\tau\mu_u^{\tau s}f_u(\mathbf{x})f_{\mathbf{z}}(\mathbf{z}|\mathbf{x},s,\tau), \quad (37)$$

which follows from (2) and Table II. Furthermore,

$$h(s,\tau,v) = cUP_D(v)o_u^v\xi_u^\tau\mu_u^{\tau s}, \quad (38)$$

$$h(\tau,v) = cUP_D(v)o_u^v\xi_u^\tau, \quad (39)$$

$$h(v) = cUP_D(v)o_u^v, \quad (40)$$

$$h[1] = cUP_D\eta^0, \quad (41)$$

where c is a constant resulting from the marginalization over $\mathbf{x}$.

For target-provided measurements, we have that $f_{\mathbf{z}}(\mathbf{z}|\mathbf{x},s,\tau) = f_{\mathbf{p}}(\mathbf{p}|\mathbf{x},s,\tau)f_\tau(\tau^{\mathbf{z}}|\tau)$. This means that

$$h(\mathbf{y}) = Uo_u^v\xi_u^\tau\mu_u^{\tau s}f_\tau(\tau^{\mathbf{z}}|\tau)f_u(\mathbf{x})f_{\mathbf{z}}(\mathbf{p}|\mathbf{x},s,\tau). \quad (42)$$

The probability of detection is omitted here, as it is defined as 1 whenever a target-provided measurement has been received. Furthermore,

$$h(s,\tau,v) = cUo_u^v\xi_u^\tau\mu_u^{\tau s}f_\tau(\tau^{\mathbf{z}}|\tau), \quad (43)$$

$$h(\tau) = cU\xi_u^\tau f_\tau(\tau^{\mathbf{z}}|\tau), \quad (44)$$

$$h(v) = cUo_u^v\sum_{\tilde{\tau}}\xi_u^{\tilde{\tau}}f_\tau(\tau^{\mathbf{z}}|\tilde{\tau}), \quad (45)$$

$$h[1] = cU\sum_{\tilde{\tau}}\xi_u^{\tilde{\tau}}f_\tau(\tau^{\mathbf{z}}|\tilde{\tau}), \quad (46)$$

where c again is a constant.

Inserting these expressions in (14)–(17) give (33)–(36), i.e., the distributions of the individual hybrid states of a new target. Furthermore, we have from Table II that g is the clutter density, which is λ for radar measurements, and 0 for target-provided measurements. We insert g and $h[1]$ in (10) and (11) to get (31) and (32). The expression for the ID probability in the event of initialization on a transmitted measurement requires some further explanation. Keeping in mind the prior distribution for the IDs (4), we have that

$$\begin{aligned}\xi_k^{t\tau j} &= \frac{h(\tau)}{h[1]} \\ &= \frac{\xi_u^\tau f_\tau(\tau^{\mathbf{z}}|\tau)}{\sum_{\tilde{\tau}}\xi_u^{\tilde{\tau}}f_\tau(\tau^{\mathbf{z}}|\tilde{\tau})} \\ &= \begin{cases} \dfrac{f_\tau(\tau^{\mathbf{z}}|\tau)(1-\xi_u^0)/|\mathcal{V}-1|}{\sum_{\tilde{\tau}}f_\tau(\tau^{\mathbf{z}}|\tilde{\tau})(1-\xi_u^0)/|\mathcal{V}-1|} & \text{if } \tau > 0 \\ 0 & \text{if } \tau = 0 \end{cases} \\ &= \begin{cases} \dfrac{f_\tau(\tau^{\mathbf{z}}|\tau)}{\sum_{\tilde{\tau}}f_\tau(\tau^{\mathbf{z}}|\tilde{\tau})} & \text{if } \tau > 0 \\ 0 & \text{if } \tau = 0 \end{cases} \\ &= \begin{cases} f_\tau(\tau^{\mathbf{z}}|\tau) & \text{if } \tau > 0 \\ 0 & \text{if } \tau = 0 \end{cases} = f_\tau(\tau^{\mathbf{z}}|\tau), \end{aligned} \quad (47)$$

where we have used that $\sum_{\tilde{\tau}}f_\tau(\tau^{\mathbf{z}}|\tilde{\tau}) = 1$. If a different prior distribution than (4) is used for the IDs, it can be accommodated by replacing the final expression with the one in the second line of the above expression. □

Proposition 3. *Updating based on a missed detection is done as*

$$w_k^{t0} = \begin{cases} 1 - r_{k|k-1}^t\eta_{k|k-1}^tP_D & \text{for radar} \\ 1 & \text{for target-provided} \end{cases}, \quad (48)$$

$$r_k^{t0} = \begin{cases} \dfrac{r_{k|k-1}^t(1-\eta_{k|k-1}^tP_D)}{1 - r_{k|k-1}^t\eta_{k|k-1}^tP_D} & \text{for radar} \\ r_{k|k-1}^t & \text{for target-provided} \end{cases}, \quad (49)$$

$$\eta_k^{t0} = \begin{cases} \dfrac{(1-P_D)\eta_{k|k-1}^t}{1 - P_D\eta_{k|k-1}^t} & \text{for radar} \\ \eta_{k|k-1}^t & \text{for target-provided} \end{cases}, \quad (50)$$

$$\xi_k^{t\tau 0} = f_{k|k-1}^t(\tau), \quad (51)$$

$$\mu_k^{t\tau s0} = f_{k|k-1}^t(s|\tau), \quad (52)$$

$$f_k^{t0}(\mathbf{x}|\tau,s) = f_{k|k-1}^t(\mathbf{x}|\tau,s). \quad (53)$$

Remark 2. The inclusion of target-provided measurement types in the case of a missed detection is somewhat artificial. The expressions are the same as for the prediction, as the absence of target-provided measurements gives no additional information to the tracking algorithm. This follows from the definition of the detection probability in Section III, i.e., that $P_D = 0$ for target-provided measurements when they have not been received. For later use, the expressions are nevertheless written out here.

Proof. We have that

$$h(\mathbf{y}) = r^t_{k|k-1}(1 - P_D(v))o^{tv}_{k|k-1}\xi^{t\tau}_{k|k-1}\mu^{t\tau s}_{k|k-1} f_{k|k-1}(\mathbf{x}|s, \tau), \tag{54}$$

where the corresponding expression from Table II has been written out. Similarly, as to what was done previously, we find through marginalization that

$$\begin{aligned} h(s, \tau, v) &= r^t_{k|k-1}(1 - P_D(v))o^{tv}_{k|k-1}\xi^{t\tau}_{k|k-1}\mu^{t\tau s}_{k|k-1} \\ h(\tau, v) &= r^t_{k|k-1}(1 - P_D(v))o^{tv}_{k|k-1}\xi^{t\tau}_{k|k-1} \\ h(v) &= r^t_{k|k-1}(1 - P_D(v))o^{tv}_{k|k-1}. \end{aligned} \tag{55}$$

Again, the different detection probabilities have to be taken into account when summing over the visibility states, giving

$$\begin{aligned} h[1] &= r^t_{k|k-1}((1 - P_D)\eta^t_{k|k-1} + (1 - \eta^t_{k|k-1})) \\ &= r^t_{k|k-1}(1 - P_D\eta^t_{k|k-1}) \end{aligned} \tag{56}$$

for radar updates and $h[1] = 1$ for AIS updates. Inserting this in (14)–(17) gives the wanted expressions for the hybrid states. Furthermore, we get from Table II that g is given by $1 - r^t_{k|k-1}$, which together with $h[1]$ gives us (48) and (49) by using (10) and (11). □

Proposition 4. *Updating based on a detection is done as*

$$w^{tj}_k = \begin{cases} P_D\eta^t_{k|k-1}r^t_{k|k-1}\sum_{\tilde\tau}\xi^{t\tilde\tau}_{k|k-1}\sum_{\tilde s}\mu^{t\tilde\tau\tilde s}_{k|k-1}l^{t\tilde\tau\tilde s j} \\ \quad \textit{for radar} \\ r^t_{k|k-1}\sum_{\tilde\tau}\xi^{t\tilde\tau}_{k|k-1}\sum_{\tilde s}\mu^{t\tilde\tau\tilde s}_{k|k-1}l^{t\tilde\tau\tilde s j} \\ \quad \textit{for target-provided} \end{cases}, \tag{57}$$

$$r^{tj}_k = 1, \tag{58}$$

$$\eta^{tj}_k = \begin{cases} 1 & \textit{for radar} \\ \eta^t_{k|k-1} & \textit{for target-provided} \end{cases}, \tag{59}$$

$$\xi^{t\tau j}_k = \frac{\xi^{t\tau}_{k|k-1}\sum_{\tilde s}l^{t\tau\tilde s j}}{\sum_{\tilde\tau}\xi^{t\tilde\tau}_{k|k-1}\sum_{\tilde s}l^{t\tilde\tau\tilde s j}}, \tag{60}$$

$$\mu^{t\tau s j}_k = \frac{\mu^{t\tau s}_{k|k-1}l^{t\tau s j}}{\sum_{\tilde s}\mu^{t\tau\tilde s}_{k|k-1}l^{t\tau\tilde s j}}, \tag{61}$$

$$f^{tj}_k(\mathbf{x}|\tau, s) = \frac{f_\mathbf{z}(\mathbf{z}|\mathbf{x}, \tau, s)f^t_{k|k-1}(\mathbf{x}|\tau, s)}{l^{t\tau s j}}, \tag{62}$$

where

$$l^{t\tau s j} = f_\tau(\tau^j|\tau)\int f_\mathbf{z}(\mathbf{z}^j_k|\tilde{\mathbf{x}})f^{t\tau s}_{k|k-1}(\tilde{\mathbf{x}})\mathrm{d}\tilde{\mathbf{x}} \tag{63}$$

for target-provided measurements and

$$l^{t\tau s j} = \int f_\mathbf{z}(\mathbf{z}^j_k|\tilde{\mathbf{x}})f^{t\tau s}_{k|k-1}(\tilde{\mathbf{x}})\mathrm{d}\tilde{\mathbf{x}} \tag{64}$$

for radar measurements.

Proof. Writing out the expression for a detection in Table II, we have that

$$\begin{aligned} h(\mathbf{y}) = r^t_{k|k-1}P_D(v)o^{tv}_{k|k-1}\xi^{t\tau}_{k|k-1}\mu^{t\tau s}_{k|k-1}\times \\ \times f^{t\tau s}_{k|k-1}(\mathbf{x})f_\mathbf{z}(\mathbf{z}|\mathbf{x}, s, \tau), \end{aligned} \tag{65}$$

which we marginalize to obtain

$$\begin{aligned} h(s, \tau, v) &= r^t_{k|k-1}P_D(v)o^{tv}_{k|k-1}\xi^{t\tau}_{k|k-1}\mu^{t\tau s}_{k|k-1}l^{t\tau s j} \\ h(\tau, v) &= r^t_{k|k-1}P_D(v)o^{tv}_{k|k-1}\xi^{t\tau}_{k|k-1}\sum_s\mu^{t\tau s}_{k|k-1}l^{t\tau s j} \\ h(v) &= r^t_{k|k-1}P_D(v)o^{tv}_{k|k-1}\sum_\tau\xi^{t\tau}_{k|k-1}\sum_s\mu^{t\tau s}_{k|k-1}l^{t\tau s j}. \end{aligned} \tag{66}$$

For radar, we have that $P_D(v = 1) = P_D$ and 0 otherwise, and for AIS $P_D(v) = P_D = 1$ if a measurement has been received. Using this, we get

$$h[1] = P_D\eta^t_{k|k-1}r^t_{k|k-1}\sum_\tau\xi^{t\tau}_{k|k-1}\sum_s\mu^{t\tau s}_{k|k-1}l^{t\tau s j} \tag{67}$$

for radar updates and

$$h[1] = r^t_{k|k-1}\sum_\tau\xi^{t\tau}_{k|k-1}\sum_s\mu^{t\tau s}_{k|k-1}l^{t\tau s j} \tag{68}$$

for AIS updates. The expressions for the hybrid states result from inserting this in (14)–(17). We see from Table II that $g = 0$, and using this together with $h[1]$, we get (57) and (58) from (10) and (11). □

D. Mixture Reduction

The mixture reduction is done similarly to what is done in the JIPDA. That is, all the association hypotheses for each track are merged. An association hypothesis $\mathbf{a}_k$ from the set of all possible association hypotheses $\mathcal{A}_k$ contains individual track-to-measurement associations a^t. The probabilities for the individual association hypotheses are

$$\Pr(\mathbf{a}_k) \propto \prod_{t \text{ s.t } a^t=0} w^{ta^t}_k \prod_{t \text{ s.t } a^t>0} w^{ta^t}_k/\lambda, \tag{69}$$

where λ is the Poisson intensity for the false alarms, and the fact that $\sum_{\mathbf{a}_k\in\mathcal{A}_k}\Pr(\mathbf{a}_k) = 1$ is used to normalize the probabilities. This in turn provides the marginal probabilities for the associations as

$$p^{tj}_k = \sum_{\mathbf{a}_k \text{ s.t. } a^t=j}\Pr(\mathbf{a}_k). \tag{70}$$

The mixture reduction remains the same irrespective of the type of measurement, as all differences are handled during the calculation of the individual posterior distributions.

Proposition 5. *We have that*

$$r_k^t = \sum_{j=0}^{m_k} r_k^{tj} p^{tj}, \tag{71}$$

$$\eta_k^t = \sum_{j=0}^{m_k} \underbrace{\frac{1}{r_k^t} r_k^{tj} p_k^{tj}}_{\beta_k^{tj}} \eta_k^{tj}, \tag{72}$$

$$\xi_k^{t\tau} = \sum_{j=0}^{m_k} \underbrace{\frac{1}{r_k^t} r_k^{tj} p_k^{tj}}_{\beta_k^{tj}} \xi_k^{t\tau j}, \tag{73}$$

$$\mu_k^{t\tau s} = \sum_{j=0}^{m_k} \underbrace{\frac{1}{\xi_k^{t\tau} r_k} \xi_k^{t\tau j} r_k^{tj} p_k^{tj}}_{\beta_k^{t\tau j}} \mu_k^{t\tau s j}, \tag{74}$$

$$f_k^{t\tau s}(\mathbf{x}) = \sum_{j=0}^{m} \underbrace{\frac{\mu_k^{t\tau s j} \xi_k^{t\tau j} r_k^{tj} p_k^{tj}}{\mu_k^{t\tau s} \xi_k^{t\tau} r_k^t}}_{\beta^{t\tau s j}} f_{k_i}^{t\tau s j}(\mathbf{x}), \tag{75}$$

where

$$\beta_k^{tj} = \frac{r_k^{tj} p_k^{tj}}{r_k^t} = \begin{cases} \dfrac{p_k^{tj}}{r_k^t}, & j > 0 \\ \dfrac{r_k^0 p_k^{t0}}{r_k^t}, & j = 0 \end{cases}, \tag{76}$$

$$\beta_k^{t\tau j} = \frac{\xi_k^{t\tau j} r_k^{tj} p_k^{tj}}{\xi_k^{t\tau} r_k^t} = \beta_k^{tj} \frac{\xi_k^{t\tau j}}{\xi_k^{t\tau}}, \tag{77}$$

$$\beta_k^{t\tau s j} = \frac{\mu_k^{t\tau s j} \xi_k^{t\tau j} r_k^{tj} p_k^{tj}}{\mu_k^{t\tau s} \xi_k^{t\tau} r_k^t} = \beta_k^{t\tau j} \frac{\mu_k^{t\tau s j}}{\mu_k^{t\tau s}}. \tag{78}$$

Using the individual $f_k^{t\tau s j}(\mathbf{x})$, the combined state $f_k^{t\tau s}(\mathbf{x})$ can be approximated by use of moment matching techniques.

Proof. The MBM containing the posterior track estimates, weights, and existence probabilities can be approximated as a multi-Bernoulli. A thorough proof of this, and more context regarding the MBM, can be found in [34]. Drawing from the aforementioned proof, in combination with the proof in [7, Appendix D], we have that the posterior distribution over $\mathbf{y}$ can be approximated as

$$f_k^t(\mathbf{y}) \approx \sum_{j=1}^{m_k} \beta_k^{tj} f_k^{tj}(\mathbf{y}) \tag{79}$$

where

$$\beta_k^{tj} = \frac{r_k^{tj} p_k^{tj}}{r_k^t} \tag{80}$$

and

$$f_k^{tj}(\mathbf{y}) = o_k^{tvj} \xi_k^{t\tau j} \mu_k^{t\tau s j} f_k^{ts\tau j}(\mathbf{x}). \tag{81}$$

Using this, together with the approximation that the visibility is independent of the other states, we can write

$$\begin{aligned}
&\sum_{j=1}^{m_k} \beta_k^{tj} f_k^{tj}(\mathbf{y}) \approx \sum_{j=1}^{m_k} \beta_k^{tj} \xi_k^{t\tau j} \mu_k^{t\tau s j} f_k^{ts\tau j}(\mathbf{x}) \sum_{j=1}^{m_k} \beta_k^{tj} o_k^{tvj} \\
&= \frac{\sum_{j=1}^{m_k} \beta_k^{tj} \xi_k^{t\tau j} \mu_k^{t\tau s j} f_k^{ts\tau j}(\mathbf{x})}{\sum_{j=1}^{m_k} \beta_k^{tj} \xi_k^{t\tau j} \mu_k^{t\tau s j}} \frac{\sum_{j=1}^{m_k} \beta_k^{tj} \xi_k^{t\tau j} \mu_k^{t\tau s j}}{\sum_{j=1}^{m_k} \beta_k^{tj} \xi_k^{t\tau j}} \times \\
&\quad \times \sum_{j=1}^{m_k} \beta_k^{tj} \xi_k^{t\tau j} \sum_{j=1}^{m_k} \beta_k^{tj} o_k^{tvj} \\
&= \sum_{j=1}^{m_k} \frac{\beta_k^{tj} \xi_k^{t\tau j} \mu_k^{t\tau s j}}{\sum_{j=1}^{m_k} \beta_k^{tj} \xi_k^{t\tau j} \mu_k^{t\tau s j}} f_k^{ts\tau j}(\mathbf{x}) \sum_{j=1}^{m_k} \frac{\beta_k^{tj} \xi_k^{t\tau j}}{\sum_{j=1}^{m_k} \beta_k^{tj} \xi_k^{t\tau j}} \mu_k^{t\tau s j} \times \\
&\quad \times \sum_{j=1}^{m_k} \beta_k^{tj} \xi_k^{t\tau j} \sum_{j=1}^{m_k} \beta_k^{tj} o_k^{tvj} \\
&= \underbrace{\sum_{j=1}^{m_k} \beta_k^{t\tau s j} f_k^{ts\tau j}(\mathbf{x})}_{f_k^{ts\tau}(\mathbf{x})} \underbrace{\sum_{j=1}^{m_k} \beta_k^{t\tau j} \mu_k^{t\tau s j}}_{\mu_k^{t\tau s}} \underbrace{\sum_{j=1}^{m_k} \beta_k^{tj} \xi_k^{t\tau j}}_{\xi_k^{t\tau}} \underbrace{\sum_{j=1}^{m_k} \beta_k^{tj} o_k^{tvj}}_{o_k^{tv}}.
\end{aligned} \tag{82}$$

Keeping in mind that $r_k^{tj} = 1 \,\forall\, j > 0$ and that $o_k^{t1j} = \eta_k^{tj} = 1 \,\forall\, j > 0$, we get the wanted expressions. Lastly, we get the expression for the existence probability r_k^t directly from [7]. □

VI. TARGET-PROVIDED MEASUREMENT HANDLING

The method shown in the previous section does not specify how the target-provided measurements are grouped before being sent to the tracker. In this section, we present three different ways of considering the target-provided measurements.

A. Method A: Sequential Measurement Processing

The first method for handling the incoming target-provided measurements is to process them, and perform the data association, as they arrive. This would mean that the predicting and updating of tracks is performed for each target-provided measurement, which can arrive at any time between radar measurement batches. This approach demands no further extensions to what is described above. The method is shown in Algorithm 1.

B. Method B: Precise Batch Measurement Processing

The second method performs the data association for the target-provided measurements at the times when radar measurements arrive. The method considers all the target-provided measurements that have arrived between the previous and current time steps as a batch of measurements. This method is conceptually similar to what is done in [13] and [17]. The method is shown

Algorithm 1 Method A: Sequential measurement processing

Require: target-provided measurements
  $Z_A = \{\mathbf{z}_A^1, \ldots, \mathbf{z}_A^m\}$, radar measurements
  $Z_R = \{\mathbf{z}_R^1, \ldots, \mathbf{z}_R^m\}$, tracks from previous time step
  $X = \{\mathbf{x}^1, \ldots, \mathbf{x}^n\}$
  for target-provided measurement $\mathbf{z}_A^j \in Z_A$ **do**
    $X \leftarrow \textsc{predict}(\mathrm{X}, \mathrm{t}_A^j)$ ▷ predict tracks to time of $\mathbf{z}_A^j$
    $X \leftarrow \textsc{update}(\mathrm{X}, \mathbf{z}_A^j)$
  end for
  $X \leftarrow \textsc{predict}(\mathrm{X}, \mathrm{t}_R)$ ▷ predict tracks to time of Z_R
  $X \leftarrow \textsc{update}(\mathrm{X}, Z_R)$

in Algorithm 2. The target-provided measurements with the same ID are clustered together, and the data association is performed based on these clusters. The clustering means that the measurement likelihood has to be calculated for each cluster rather than for each measurement. The measurement likelihood for I_m measurements with the same ID is

$$f_{\mathbf{z}}(\mathbf{z}|\mathbf{x}) = f_{\mathbf{z}}(\mathbf{z}^1, \ldots, \mathbf{z}^{I_m}|\mathbf{x}) = \prod_{i=1}^{I_m} f_{\mathbf{z}}(\mathbf{z}^i|\mathbf{z}^{i-1}, \ldots, \mathbf{z}^1, \mathbf{x}), \tag{83}$$

where

$$f_{\mathbf{z}}(\mathbf{z}^i|\mathbf{z}^{i-1}, \ldots, \mathbf{z}^1, \mathbf{x}) = \int f_{\mathbf{z}}(\mathbf{z}^i|\mathbf{x}^i) f_{\mathbf{x}}(\mathbf{x}^i|\mathbf{z}^{i-1}, \ldots, \mathbf{z}^1, \mathbf{x}) \mathrm{d}\mathbf{x}^i. \tag{84}$$

This has to be calculated for each measurement that has arrived between the radar updates. The measurements are sorted according to their time stamp, with $\mathbf{z}^{I_m}$ being the most recent measurement. This expression effectively replaces the integral in (63). The individual kinematic states are calculated as

$$f_k^{t\tau sj}(\mathbf{x}|\mathbf{z}^i, \mathbf{z}^{i-1}, \ldots, \mathbf{z}^1, \mathbf{x}) = \frac{f_{\mathbf{z}}(\mathbf{z}^i|\mathbf{x}^i) f_{\mathbf{x}}(\mathbf{x}^i|\mathbf{z}^{i-1}, \ldots, \mathbf{z}^1, \mathbf{x})}{\int f_{\mathbf{z}}(\mathbf{z}^i|\mathbf{x}^i) f_{\mathbf{x}}(\mathbf{x}^i|\mathbf{z}^{i-1}, \ldots, \mathbf{z}^1, \mathbf{x}) \mathrm{d}\mathbf{x}^i}. \tag{85}$$

This expression can be calculated using, e.g., a Kalman filter. A thorough explanation of this recursive measurement likelihood calculation can be found in the supplementary material of [13]. With these expressions established, the other calculations and expressions are identical to Method A.

C. Method C: Batch Measurement Processing With Added Noise

In Section III, it is assumed that the radar measurements of a single measurement batch are synchronized, i.e., they all arrive at the same time. We do not make the same assumption for the target-provided measurements. However, making this assumption would allow us to simplify the handling of the measurements and remove some of the computational complexity of the above methods. Such an approach would be well suited when the radar frequency is high, as the timing errors would be small. Algorithm 3 describes the approach. Furthermore, only the most recent measurement is considered when a target has transmitted more than one measurement between radar updates. In addition, this method should be used with a higher measurement noise level to account for the synchronization errors.

Algorithm 2 Method B: Precise batch measurement processing

Require: target-provided measurement clusters
  $Z_A = \{\mathbf{z}_A^1, \ldots, \mathbf{z}_A^m\}$, radar measurements
  $Z_R = \{\mathbf{z}_R^1, \ldots, \mathbf{z}_R^m\}$, tracks from previous time step
  $X = \{\mathbf{x}^1, \ldots, \mathbf{x}^n\}$
  for track $\mathbf{x}^t \in X$ **do**
    for target-provided measurement cluster $\mathbf{z}_A^j \in Z_A$ **do**
      $\mathbf{x}^{t,j} \leftarrow \textsc{copy}(\mathbf{x}^t)$
      for target-provided measurement $\mathbf{z}^i \in \mathbf{z}_A^j$ **do**
        $\mathbf{x}^{t,j} \leftarrow \textsc{predict}(\mathbf{x}^t, \mathrm{t}_A^{j,i})$
        $\mathbf{x}^{t,j} \leftarrow \textsc{update}(\mathbf{x}^j, \mathrm{t}_A^{j,i})$
      end for
      $l^{t,j} \leftarrow \textsc{measurementLikelihood}(\mathbf{x}^{t,j}, \mathbf{z}_A^j)$
      $\mathbf{x}^{t,j} \leftarrow \textsc{predict}(\mathbf{x}^{t,j}, \mathrm{t}_R)$
    end for
    $X_{\mathrm{new}}^{t,j} \leftarrow \mathbf{x}^{t,j}$
  end for
  $X \leftarrow \textsc{mixtureReduction}(X_{\mathrm{new}}, l)$
  $X \leftarrow \textsc{update}(\mathrm{X}, Z_R)$

Algorithm 3 Method C: Batch measurement processing with added noise

Require: target-provided measurements
  $Z_A = \{\mathbf{z}_A^1, \ldots, \mathbf{z}_A^m\}$, radar measurements
  $Z_R = \{\mathbf{z}_R^1, \ldots, \mathbf{z}_R^m\}$, tracks from previous time step
  $X = \{\mathbf{x}^1, \ldots, \mathbf{x}^n\}$
$X \leftarrow \textsc{predict}(\mathrm{X}, \mathrm{t}_R)$ ▷ predict tracks to time of Z_R, Z_A
$X \leftarrow \textsc{update}(\mathrm{X}, Z_A)$
$X \leftarrow \textsc{update}(\mathrm{X}, Z_R)$

Remark 3. When grouping the same-ID target-provided measurements, one has to keep in mind the assumption of only one measurement arising from each target. If a target transmits two target-provided measurements between radar updates, and one of the measurements has a corrupted ID number, then this would breach the assumption. The most obvious way to amend this is to discard target-provided measurements whenever there are more measurements than tracks present. This will, however, interfere with initializing new tracks on the target-provided measurements. It should also be noted that if the radar frequency is higher than the target-provided measurement transmission frequency, a cluster will always only contain a single measurement. This would avoid the aforementioned problem and simplify calculations.

Remark 4. When using (83), the discrete hybrid states will take their most likely value as a mean over the information from the measurements in the cluster. This is as opposed to obtaining the most likely value at the most recent target-provided measurement. This could theoretically impact the estimation of the discrete states. For example, if two measurements in a cluster indicate two different kinematic models, then this disparity will not be captured when using the batch processing methods.

VII. IMPLEMENTATION

A. Utilization of Gaussian-Linearity

To make the implementation tractable, we model the individual kinematic states and the measurement likelihoods as Gaussian distributions. This allows us to use an Extended Kalman Filter when predicting and updating the kinematic estimates. The measurement likelihoods are defined as

$$f^R_{\mathbf{z}}(\mathbf{z}_k|\mathbf{y}_k) = \mathcal{N}(\mathbf{z}_k|\mathbf{H}_R\mathbf{x}, \mathbf{R}_R) \tag{86}$$

for radar measurements and as

$$f_{\mathbf{p}}(\mathbf{p}_k|\mathbf{y}_k) = \mathcal{N}(\mathbf{p}_k|\mathbf{H}_A\mathbf{x}, \mathbf{R}_A) \tag{87}$$

for the positional part of the AIS measurements. Furthermore, the kinematic transition density $f^{s\tau}_{\mathbf{x}}(\mathbf{x}_k|\mathbf{x}_{k-1})$ is assumed to be in the form of a Gaussian

$$f^{s\tau}_{\mathbf{x}}(\mathbf{x}_k|\mathbf{x}_{k-1}) = \mathcal{N}(\mathbf{x}_k|\mathbf{f}^{(s)}(\mathbf{x}_{k-1}), \mathbf{Q}^{(s)}). \tag{88}$$

The transition model is linearized when needed to enable EKF prediction and Gaussian moment matching for mixture reduction.

The kinematic unknown target density from (2) is defined as

$$f_u(\mathbf{x}) = 1_\Omega(\mathbf{H}^{(s)}\mathbf{x})\mathcal{N}(\mathbf{H}^{*(s)}\mathbf{x}; \mathbf{0}, \mathbf{P}_v), \tag{89}$$

where $1_\Omega(\cdot)$ is an indicator function, which is 1 when the unknown target is within the surveillance area, and $\mathbf{H}^{*(s)}$ is the permutation matrix corresponding to the nonpositional states of the state vector $\mathbf{x}$. Using this, we have that

$$\begin{aligned} &f_{\mathbf{z}}(\mathbf{z}|\mathbf{x}, s, \tau) f_u(\mathbf{x}) \\ &= 1_\Omega(\mathbf{H}^{(s)}\mathbf{x})\mathcal{N}(\mathbf{z}^{a^t}_k|\mathbf{H}^{(s)}\mathbf{x}, \mathbf{R}^s)\mathcal{N}(\mathbf{H}^{*(s)}\mathbf{x}|\mathbf{0}, \mathbf{P}^{(s)}_v). \end{aligned} \tag{90}$$

In the case of a large enough surveillance area Ω, and under the assumption of Gaussian-linearity, this can be approximated as $\mathcal{N}(\mathbf{x}|\hat{\mathbf{x}}^s_0, \mathbf{P}^s_0)$. Furthermore, this means that the constant c in Proposition 2 becomes

$$c = \int f_{\mathbf{z}}(\mathbf{z}|\mathbf{x}, s, \tau) f_u(\mathbf{x})\mathrm{d}\mathbf{x} \approx \int \mathcal{N}(\mathbf{x}|\hat{\mathbf{x}}^s_0, \mathbf{P}^s_0)\mathrm{d}\mathbf{x} = 1. \tag{91}$$

A more thorough proof regarding the unknown target density can be found in Appendix C of [7].

B. Gating

Because the target-provided measurements can arrive at any time, the number of times we have to perform gating increases considerably. The main computational cost of this is the number of predictions. Thus, we should consider this when creating the gating procedure.

Several different gating methods are presented in [33]. The first method relies on gating for each kinematic model, and it uses all measurements that have been gated by any of the models. A different method is a centralized gating procedure, which makes an approximation across all models using a single gate. We use a somewhat more refined method, the Two-Step Model Probability Weighted Gating (TS-MPWG) method. TS-MPWG was also presented in [33]. The first step in the method is a centralized gating procedure

$$f^t_{k|k-1}(\mathbf{x}) = \sum_{\tilde{\tau}} \xi^{t\tilde{\tau}}_{k|k-1} \sum_{\tilde{s}} \mu^{t\tilde{\tau}\tilde{s}}_{k|k-1} f^{t\tilde{\tau}\tilde{s}}_{k|k-1}(\mathbf{x}), \tag{92}$$

where $f^t_{k|k-1}(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\hat{\mathbf{x}}_{k|k-1}, \hat{\mathbf{P}}_{k|k-1})$ provides the gate center $\hat{\mathbf{x}}_{k|k-1}$ and the predicted covariance $\hat{\mathbf{P}}_{k|k-1}$. Furthermore, the innovation covariance becomes

$$\mathbf{S} = \mathbf{H}\hat{\mathbf{P}}_{k|k-1}\mathbf{H}^\top + \mathbf{R}_k. \tag{93}$$

If no measurements are gated during the first step, then the next step is initiated. Here, the gate is determined by the largest possible model error and should encompass any measurements generated by the target even if the chosen kinematic model is wrong. Thus, the TS-MPWG method can exploit the more computationally effective nature of the central gating method while compensating for eventual model errors. Adapting the expressions in [33] to this model, the gate in the second step is determined by the maximal difference between $\hat{\mathbf{x}}_{k|k-1}$ and the individual $\hat{\mathbf{x}}^{t\tau s}_{k|k-1}$. This error is

$$K_{\max} = \arg\max_{\tau,s} \|\mathbf{H}\hat{\mathbf{x}}_{k|k-1} - \mathbf{H}\hat{\mathbf{x}}^{t\tau s}_{k|k-1}\|^2. \tag{94}$$

Using this, we calculate the gate volume as

$$\mathbf{S}_d = \mathbf{S} + \mathbf{K}_{\max} \tag{95}$$

where

$$\mathbf{K}_{\max} = \mathrm{diag}[\overbrace{K_{\max}, \ldots, K_{\max}}^{n}] \tag{96}$$

for a measurement space of dimension n.

Furthermore, it would be beneficial to have the possibility of gating target-provided measurements between two radar time steps without having to predict the state of all tracks. We can achieve this by utilizing one of the methods described in [36]. The method involves expanding the gate size according to a fixed presumed maximum velocity. That is, rather than predicting the track from time t_{k-1} to t_k, the gate accounts for movement in all directions at a very high speed. This method gives very large validation gates, and we only use it as a preliminary step before using the TS-MPWG method. Here, the radius of the gate is decided by

$$r_k = 2r_{k_0} + (t_k - t_{k-1})v_{\max} \tag{97}$$

where $v_{\max}$ is a parameter representing the largest possible speed for a target, and

$$r_{k_0} = \sqrt{\gamma_G \text{eig}(\mathbf{R})_{\max}}. \tag{98}$$

Here, γ is the gate size, and $\text{eig}(\mathbf{R})_{\max}$ is the largest eigenvalue of the measurement covariance matrix.

C. Initialization and Termination

Due to target-provided measurements never being clutter measurements, care should be taken when choosing the initialization scheme. In JIPDA tracking algorithms, new tracks are usually only initialized on so-called free measurements, i.e., measurements that have not been gated by any tracks at the current time step. When using this scheme, a target-provided measurement belonging to an uninitialized target, which falls within the validation gate of a previously initialized target, would most likely assign the measurement to the previously initialized target. However, a scheme that initiates tracks on all measurements will avoid this problem.

Initializing a new track on every measurement is computationally expensive and requires measures to mitigate computational complexity. For this purpose, we classify the tracks as newborn, adolescent, and ordinary. Newborn tracks are tracks that have been initialized at the current time step, adolescent tracks are tracks that were initialized at the previous time step, and ordinary tracks are all other tracks. The adolescent tracks are not allowed to compete for measurements in the same way as the ordinary tracks. The restriction comes into play when an adolescent track i and an ordinary track t have gated measurement j at the current time step, and they have both gated the same measurement at the previous time step. Then, the adolescent track j is only allowed to compete for the measurement if it has a larger weight relative to the measurement than the other track

$$\max_{t,j} w_k^{tj} < T_B w_k^{ij}, \tag{99}$$

where T_B is a threshold parameter. Otherwise, the adolescent track is not allowed to compete for measurement j, which is enforced by setting $w_k^{ij} = 0$.

Termination is done as described in [37]. First, any tracks with an existence probability under a predetermined threshold T_d are removed. Furthermore, any two tracks deemed to be identical are identified by the use of the hypothesis test in [1, p. 447]. The most recently initialized of these are then terminated. Lastly, any tracks that have not been associated with a measurement for N_T radar intervals are terminated.

D. Kinematic Models

The implementation uses two different kinematic models: the constant velocity (CV) model and the coordinated turn (CT) model. Due to the varying prediction intervals, we use the discretized continuous formulation of the models. The CV model has the kinematic state $\mathbf{x} = [x, y, v_x, v_y]^{\mathrm{T}}$ where v denotes the velocity, and the state evolves according to $\mathbf{x}_k = \mathbf{F}^{(s)}(\Delta t)\mathbf{x}_{k-1} + \mathbf{v}_k$, $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}^{(s)})$ where

$$\mathbf{F}^{(s)} = \begin{bmatrix} \mathbf{I}_2 & \Delta t \mathbf{I}_2 \\ \mathbf{0} & \mathbf{I}_2 \end{bmatrix}, \ \mathbf{Q}^{(s)} = \begin{bmatrix} (\Delta t)^3/3\mathbf{I}_2 & (\Delta t)^2/2\mathbf{I}_2 \\ (\Delta t)^2/2\mathbf{I}_2 & \Delta t \mathbf{I}_2 \end{bmatrix} q. \tag{100}$$

Here, $\mathbf{I}$ is the identity matrix, Δt is the prediction interval, and q is the process noise intensity [2, p. 270] of the process noise. The CT model has an additional state ω, which is the turn rate. It evolves as $\mathbf{x}_k = \mathbf{F}^{(s)}(\mathbf{x}_{k-1})\mathbf{x}_{k-1} + \mathbf{v}_k$, $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}^{(s)})$ where

$$\mathbf{F}^{(s)}(\mathrm{x}) = \begin{bmatrix} 1 & 0 & \frac{\sin \Delta t\omega}{\omega} & \frac{-1 + \cos \Delta t\omega}{\omega} & 0 \\ 0 & 1 & \frac{1 - \cos \Delta t\omega}{\omega} & \frac{\sin \Delta t\omega}{\omega} & 0 \\ 0 & 0 & \cos \Delta t\omega & -\sin \Delta t\omega & 0 \\ 0 & 0 & \sin \Delta t\omega & \cos \Delta t\omega & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{101}$$

and

$$\mathbf{Q}^{(s)} = \begin{bmatrix} \mathbf{Q}^{(1)} & \mathbf{0} \\ \mathbf{0} & \Delta t q_\omega \end{bmatrix}, \tag{102}$$

where $\mathbf{Q}^{(1)}$ is a CV model covariance matrix and q_ω is the intensity of the turn rate process noise. In the implementation, the CT model is linearized as in [2, Sec. 11.7.2].

Remark 5. In most IMM applications, the transition matrix is constant. Thus, an aspect that has to be considered when the measurements do not arrive at a fixed frequency, is how to design the time-varying transition matrix $\Pi(\Delta t)$. A solution is to use the theory of Continuous Markov Chains to get an approximation for $\Pi(\Delta t)$ from the time-independent transition matrix Π. As described in [16], this can be done by use of a generator matrix G. The generator matrix is closely related to the time-independent transition matrix Π and is formulated as

(a) no transition takes place in the time interval Δt with probability $1 + g_{ii}\Delta t + o(\Delta t)$,
(b) a transition takes place in the time interval Δt with probability $g_{ij}\Delta t + o(\Delta t)$,

where g_{ij} are the individual elements of G and $o(\Delta t)$ indicates some small additional term, which is ignored.

This approximation is reasonable for relatively small Δt. Thus, the generator matrix G for M number of states can be written as

$$G = \begin{bmatrix} \pi^{11} - 1 & \dots & \pi^{1M} \\ \vdots & \ddots & \vdots \\ \pi^{M1} & \dots & \pi^{MM} - 1 \end{bmatrix}, \tag{103}$$

where π^{ij} are the individual elements of Π. Furthermore, we have from [16] that

$$\pi^{ij}(\Delta t) \approx g_{ij}\Delta t \text{ if } i \neq j \text{ and } \pi^{ii}(\Delta t) \approx 1 + g_{ii}\Delta t. \tag{104}$$

Using this, we get

$$\Pi(\Delta t) \approx \begin{bmatrix} 1 + (\pi^{11} - 1)\Delta t & \dots & \pi^{1M}\Delta t \\ \vdots & \ddots & \vdots \\ \pi^{M1}\Delta t & \dots & 1 + (\pi^{MM} - 1)\Delta t \end{bmatrix}. \tag{105}$$

E. Measurement Models

Radar Measurements

The radar measurements only contain positional data, and the measurements can be written as

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{w}_k, \quad \mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_R). \tag{106}$$

The noise matrix has both a Cartesian and polar element, to account both for errors in range and bearing, and clustering errors. The measurement noise matrix for the radar measurement becomes

$$\mathbf{R}_R = \mathbf{R}_c + \mathbf{R}_p. \tag{107}$$

Here, $\mathbf{R}_c$ is the Cartesian noise component, while $\mathbf{R}_p$ is the polar noise component converted to Cartesian coordinates. The conversion is done by using the unbiased conversion equations from [22].

Target-Provided Measurements

The target-provided measurements can contain both positional and velocity data. The kinematic part of the measurements can be written as

$$\mathbf{p}_k = \mathbf{H}^{\text{pos}}\mathbf{x}_k + \mathbf{H}^{\text{vel}}\mathbf{x}_k + \mathbf{w}_k, \quad \mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_A), \tag{108}$$

where $\mathbf{H}^{\text{pos}}$ and $\mathbf{H}^{\text{vel}}$ are the position and velocity measurement matrices, respectively. The position is usually derived from GPS information, while the velocity is derived either from a combination of speed and heading data [6]. Due to the nature of the data, we approximate the positional errors as Cartesian noise, while we approximate the velocity errors as polar noise. The measurement noise matrix for the AIS measurement becomes

$$\mathbf{R}_A = \mathbf{H}^{\text{pos}}\mathbf{R}_{c,A} + \mathbf{H}^{\text{vel}}\mathbf{R}_{p,A}, \tag{109}$$

where $\mathbf{R}_{c,A}$ is the Cartesian noise component, while $\mathbf{R}_{p,A}$ is the polar noise component converted to Cartesian coordinates, again by using [22].

Table III
Tracking System Parameters

Quantity	Symbol	unit	Value
Radar sample interval	T	[s]	2.5
Model 1 process noise intensity	$q_{a,1}$	[m^{2}s^{-3}]	0.1^2
Model 2 process noise intensity	$q_{a,2}$	[m^{2}s^{-3}]	1.5^2
Turn rate process noise intensity	q_ω	[rad^{2}s^{-3}]	0.02^2
Cartesian noise std. radar	σ_{c_R}	[m]	6.6
Cartesian noise std. AIS	σ_{c_A}	[m]	3.0
Polar range std.	σ_r	[m]	8.0
Polar bearing std.	σ_θ	[°]	1.0
Detection probability	P_{D}	[%]	92
Survival probability	P_{S}	[%]	99.9
Noncorrupted ID probability	P_C	[%]	99
Initial visibility probability	η_u	[%]	90
Visibility Markov probability	ω^{11}	[−]	0.90
Visibility Markov probability	ω^{10}	[−]	0.10
Visibility Markov probability	ω^{01}	[−]	0.52
Visibility Markov probability	ω^{00}	[−]	0.48
Gate size	γ	[−]	3.5
Clutter intensity	λ	[m^{-2}]	5×10^{-7}
Unknown target rate	U	[m^{-2}]	5×10^{-8}
Initial velocity std.	σ_v	[m s^{-1}]	10
Initial model probability	μ_u^s	[%]	[80 10 10]
Unknown target no ID probability	ξ_u^0	[−]	0.5
Existence confirmation threshold	T_c	[%]	99.9
Existence termination threshold	T_d	[%]	1
IMM transition probability	$\pi^{\bar{s}s}$	[%]	$\begin{bmatrix} 99 & .5 & .5 \\ .5 & 99 & .5 \\ .5 & .5 & 99 \end{bmatrix}$

VIII. RESULTS

A. Simulation Environment

We created the simulated data in line with the assumptions in Section III. The ownship is situated at the origin and is stationary. The surveillance area is circular with a radius of 500 m. We track five targets, all appearing at the edge of the area. Three of the targets appear at time $t = 0$ s, while the last two appear at time $t = 10$ s. The data consists of true target positions, radar, and AIS measurements. The movement of the targets follows a CV model with process noise intensity $q = 0.1^2\text{m}^2\text{s}^{-3}$, with occasional maneuvers according to a CT model. Furthermore, all targets are guided toward the center of the surveillance area until they are within 50 m of it. The measurements are created according to the measurement models in Section VII-E.

The tracking parameters were tuned to achieve good performance on experimental data and are similar to the ones in [7]. We list the parameters in Table III. These are also the parameters used for creating the simulated data. The AIS measurement noise was also chosen according to the experimental data and would correspond to the measurements providing high location accuracy. Furthermore, in practical applications, the precision of the AIS location data can be dynamically adjusted according to a position accuracy flag in the AIS protocol [19].

To evaluate the results, we used five different performance measures: the optimal subpattern assignment (OSPA) metric [28], the track localization error (TLE), track fragmentation rate (TFR), track false alarm rate (TFAR), and track probability of detection (TPD). The last four evaluation methods are described in [26]. The OSPA metric provides an overall performance assessment, while the other measures provide information about specific aspects of the methods.

We tested five different methods: The three methods described in Section VI, a method using only the radar measurements, and the method described by Gaglione *et al.* in [13]. The method from [13] uses a particle filter and loopy belief propagation and is thus very different from the one described in this paper. We denote the method from [13] as the belief propagation and particle filter method (BP-PF method). The implementation uses a single CV model with process noise intensity $q = 0.7^2\ \mathrm{m^2 s^{-3}}$, and the same parameters as in Table III where applicable. As proposed in [23], of which the method in [13] is an extension, we use 3000 particles for each potential target. We set the number of potential targets to 30, as is done in [13].

The code implementing Method A from Section VI is available at [18].

B. Simulated Data

We tested the methods on 100 simulated data sets over a range of different detection probabilities. The results are seen in Figures 2 and 3. Not surprisingly, the pure radar tracking method performs worse than the AIS-aided tracking methods from Section VI when the P_D is low. The difference becomes smaller as P_D approaches 1, but is still significant. Furthermore, we see that the method from [13] generally performs worse than all the methods in Section VI, and, in some aspects, worse than the pure radar tracking method. The right subfigure in Fig. 2 shows that the largest difference in performance is in the initial stage of the scenarios. That is, the method from [13] struggles with initialization relative to the other methods. This struggle to initialize tracks also results in significantly worse TPD, whereas the other methods perform similarly to each other.

Furthermore, the TLE of the method from [13] is better than that of the pure radar method, but it is still worse than the other methods. We see that the three methods from Section VI perform similarly. As expected, the batch processing method using added noise gives slightly less precise estimates. While we see some differences between the methods for TFR and TFAR, the errors are of an overall small magnitude. However, the pure radar tracker is more prone to track fragmentation than the other methods.

The computational complexity of the methods also warrants a comparison. The pure radar tracker is the least computationally demanding, as all the other methods add functionality in addition to performing the calculations of the pure radar tracker. The precise batch processing method is the most demanding of the target-provided measurement handling methods. This is because it requires predictions and updates for each track for each measurement. The least demanding of the three is the batch processing method with added noise, as it does not need to perform more predictions than the pure radar methods. The three methods generally do not introduce a prohibiting amount of complexity and can all be implemented using a Kalman filter. Furthermore, they are all significantly less demanding than the BP-PF method, as it uses a particle filter.

C. Experimental Data

In addition to the simulated data, the sequential measurement handling method, the pure radar tracker, and the method from [13] were tested on experimental data collected as part of the Autosea project at NTNU [8]. The data set is the same set used in [7]. We consider two scenarios, which include three different ships using AIS,

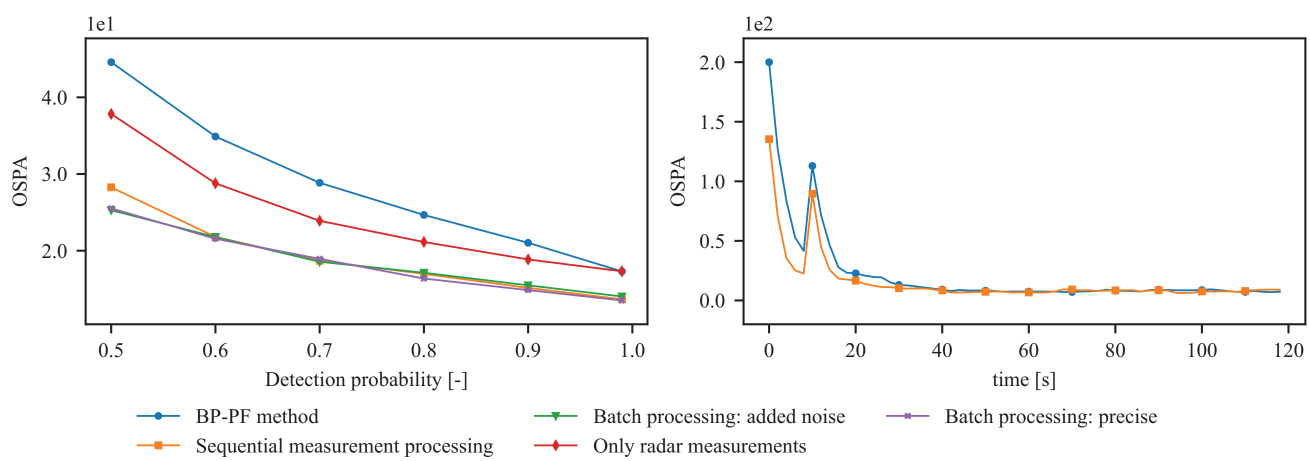


Fig. 2. Comparison of the different methods using the OSPA metric. The left figure shows the average OSPA values of each method for different detection probabilities. The right figure shows the average OSPA value for each time step, with $P_D = 0.9$. Here, we only consider the BP-PF method and the sequential measurement processing method. Both figures contain results from the same 100 scenarios. The OSPA values are calculated using $p = 2$ and $c = 200$. The purpose of the two parameters is described in [28].

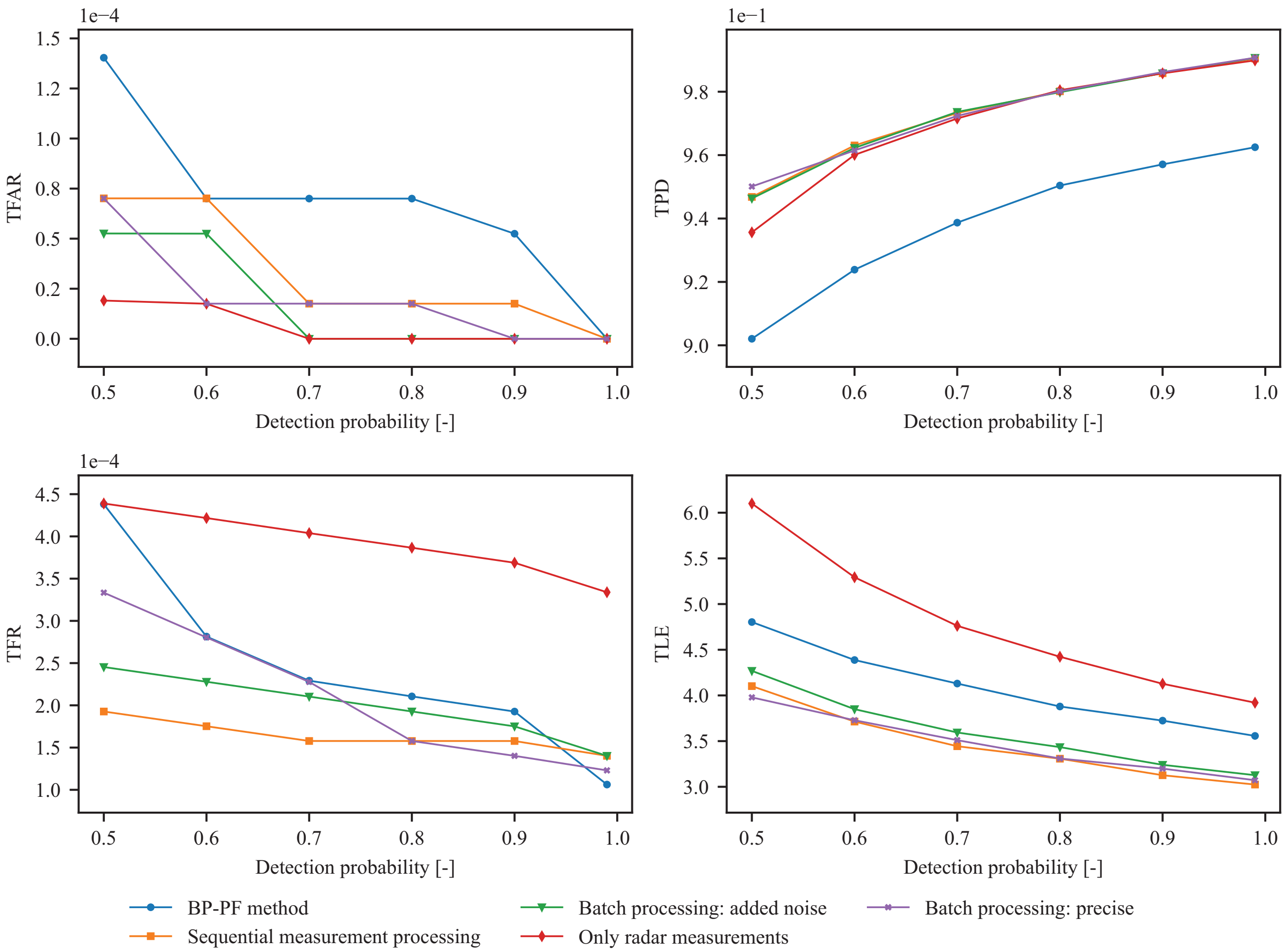


Fig. 3. TFAR, TPD, TLE, and TFR are the five different methods for different detection probabilities. The values were calculated by running the methods on the same 100 scenarios as above.

of which two provide frequent measurements. The transmission frequency for the two ships is higher than what is mandated by the IMO [19], but the data set is nevertheless helpful for demonstrating the functionality and usefulness of the tracking method. Due to the AIS data previously being used as ground truth for the AIS-equipped vessels, the AIS data has been interpolated to increase the number of measurements. This interpolation was undone prior to using the data, i.e., we removed any artificially added measurements.

Figure 4 shows the results from the first scenario. The scenario contains three fast-moving and maneuvering targets and a single slow-moving target. The slow-moving target is a large vessel with an AIS transmitter, while the three fast-moving targets are small, rigid inflatable boats (RIBs). Only one of the RIBs has an AIS transmitter, and it only transmits a single AIS measurement. The large vessel, however, provides high-quality AIS measurements. As can be seen, both the sequential measurement handling method and the pure radar method can track the scenario well, while the BP-PF method struggles. The BP-PF method likely struggles due to the kinematic modeling, i.e., because it has to use a single model to cover the kinematic behavior of both the RIBs and the large vessel. The two other methods have more flexibility in their use of IMM, and they can thus use different kinematic models for the RIBs and the large ship. When combining target-provided measurements with IMM, the tracker is also better able to select the correct kinematic model for each target. Furthermore, the sequential measurement handling method can use the AIS measurements when tracking the large vessel, improving upon the track from the pure radar method. It also correctly associates the single AIS measurement transmitted by the RIB.

The second scenario can be seen in Fig. 5. The plots show the two vessels with frequent AIS transmissions and the ownship. Figure 6 displays a close-up of the northernmost turn, with and without AIS measurements. The second scenario highlights some advantages of utilizing the AIS measurements when available. The main event occurs during the turn depicted in Fig. 6, where the radar measurements are poor due to the large vessel making a maneuver and generating numerous clutter measurements. A similar effect also occurs on the straight leading up to the turn. Both of these effects

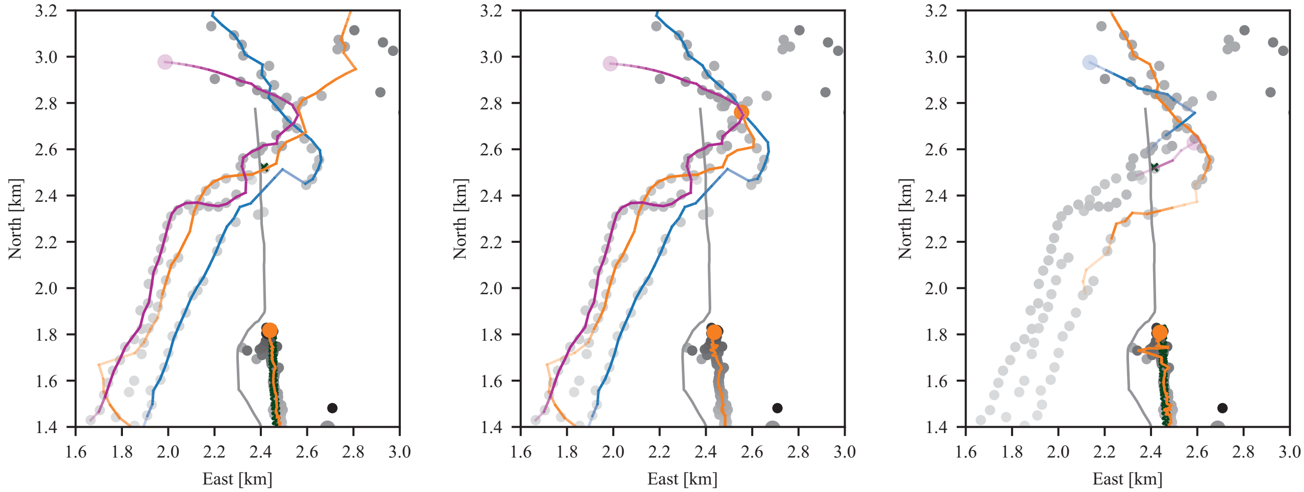


Fig. 4. A scenario showing four targets. The ownship is the gray line, moving southwards, while the targets all move northwards. The gray dots are radar measurements, and the green crosses are AIS measurements. The measurements become more transparent as time passes, i.e., the darker ones have arrived closer to the end of the scenario. The transparency of the tracks is decided by the existence probability, with the more transparent having a lower probability of existence. The target originating furthest to the right is a large vessel with an AIS transmitter, while the three other targets are small, fast-moving RIBs. Of the RIBs, only the orange has an AIS transmitter, which transmits a single measurement during the scenario. The RIBs make several maneuvers before moving beyond the radar range. (a) Results when tracking the scenario using Method A: Sequential measurement processing. (b) Results when tracking the scenario using only radar. (c) Results when tracking the scenario using the BP-PF method.

cause the purely radar-guided tracking method to veer off track, while the sequential measurement handling method can utilize the AIS measurements to avoid this. The BP-PF method loses track on the straight due to a shift in the radar measurements, combined with a temporary absence of AIS measurements, but is better able to handle the northernmost turn than the pure radar tracker. This improvement comes at the expense of a falsely initialized track on the unused radar measurements. The false track is avoided when using the sequential measurement handling method, given the correct tuning. Figure 7 shows the estimated course of the target during the turn, in addition to the standard deviation of the estimates. The poor radar measurements make the course estimates unreliable when not also utilizing the AIS measurements. When using the AIS measurements, the standard deviation of the course estimates during the turn is significant, but they are still considerably smaller than when the tracker uses only radar measurements. Furthermore, the track avoids sudden course changes. In this scenario, the inclusion of AIS measurements causes no unwanted consequences, opening the

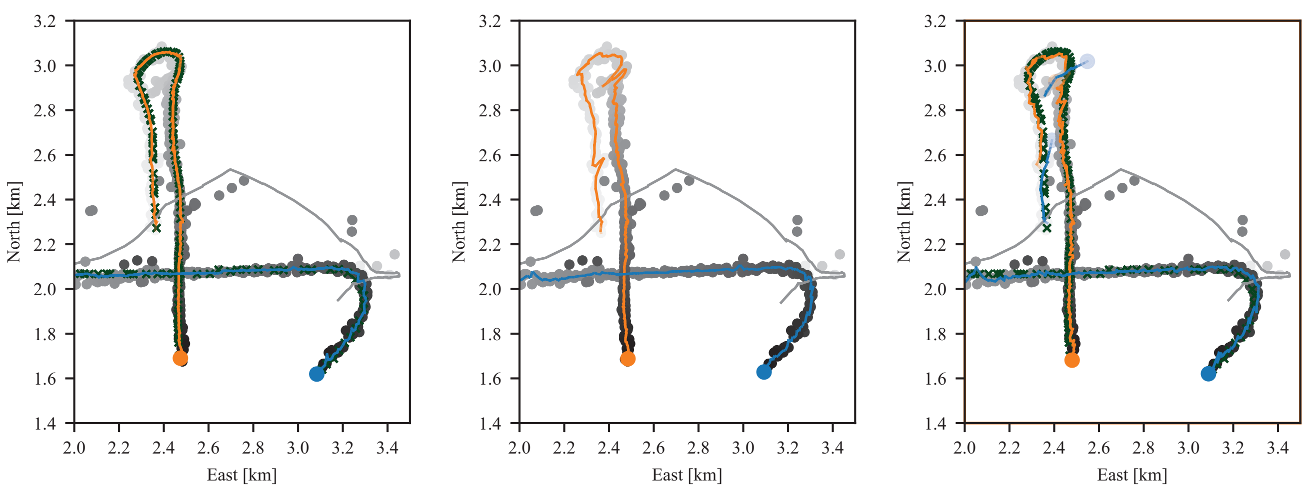


Fig. 5. A scenario showing two large vessels with AIS transmitters (with tracks shown as blue and orange lines), in addition to an ownship (gray line). We depict the measurements and tracks as in Fig. 4. Initially, the orange target moves north, while the blue target moves east. After some time, the orange target makes a u-turn, while the blue target makes a turn toward southwest. The ownship moves in a clockwise motion. The orange and blue dots represent the track positions at the end of the scenario. (a) Results when tracking the scenario using Method A: Sequential measurement processing. (b) Results when tracking the scenario using only radar. (c) Results when tracking the scenario using the BP-PF method.

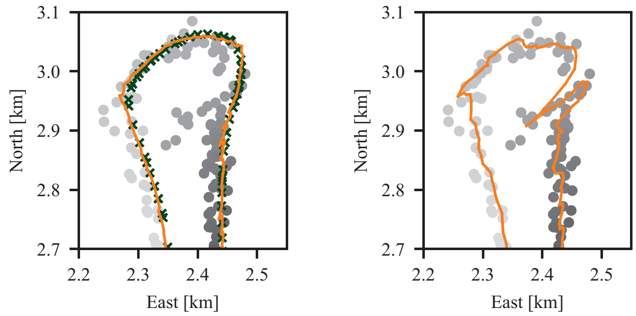


Fig. 6. A closer look at the northernmost turn for the orange track in the scenario in Figure 5. A single large vessel makes a clockwise turn, resulting in significant amounts of radar clutter. (a) A target making a clockwise turn while being tracked using AIS and radar. (b) A target making a clockwise turn while being tracked using only radar.

possibility of utilizing all the potential enhancements information given by the messages can bring.

IX. CONCLUSION

We present a framework for including target-provided measurements in a JIPDA-based tracking algorithm. We use AIS measurements as an example of such measurements. It is seen that the inclusion of such measurements can help a pure radar tracking method and improve performance greatly when the radar measurements are of low quality. In addition to the pure performance improvements, target-provided measurements can facilitate the identification of targets, which can be useful for, e.g., a collision avoidance system. Furthermore, we present and compare three different methods of handling the target-provided measurements: One method where the tracker processes the target-provided measurements when they arrive, and two methods where the tracker processes them at the time of the radar update. All three methods outperform similar state-of-the-art methods.

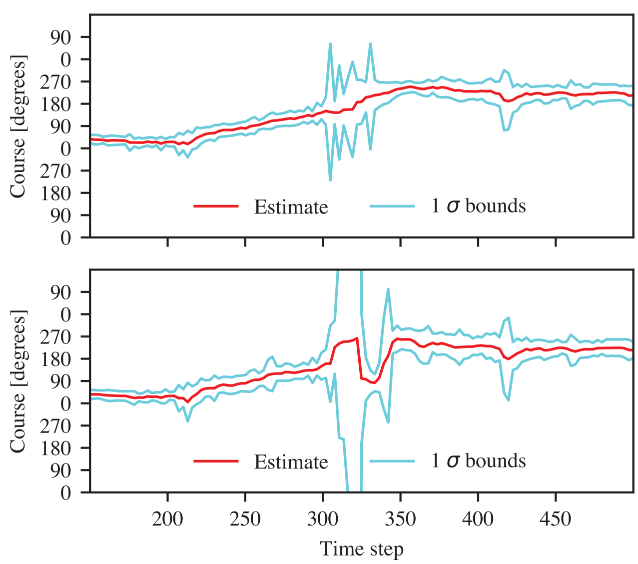


Fig. 7. Course estimate for the turn depicted in Figure 6 using both radar and AIS (top) and using only radar (bottom).

A. Future Work

The main focus of this work is how to incorporate target-provided measurements into a tracking method, and we have avoided a more thorough analysis of how to exploit the information provided by different protocols. Thus, how to use more of the data provided by such measurements should be investigated. There is also the possibility of using the expressions presented in Section V in a PMBM, which could improve performance. Another option is to use target-provided measurements to assist in clustering radar measurements. Lastly, there are safety concerns when using target-provided information. That is, the inclusion of easily manipulated input in a safety-critical system should be investigated.

ACKNOWLEDGMENT

The radar data were recorded by Erik Wilthil, Andreas Flåten, Bjørn-Olav Eriksen, and Giorgio D. K. M. Kufoalor, with assistance from Maritime Robotics and Kongsberg.

REFERENCES

[1] Y. Bar-Shalom and X. R. Li
Multitarget-Multisensor Tracking: Principles and Techniques. Storrs, CT, USA: Yaakov Bar-Shalom, 1995.

[2] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan
Estimation with Application to Tracking and Navigation. Hoboken, NJ, USA: Wiley, 2001.

[3] Y. Bar-Shalom and E. Tse
"Tracking in a cluttered environment with probabilistic data association,"
Automatica, vol. 11, no. 5, pp. 451–460 1975.

[4] O. Baud, N. Honore, and O. Taupin
"Radar/ADS-B data fusion architecture for experimentation purpose,"
in *Proc. 9th Int. Conf. Inf. Fusion*, 2006, pp. 1–6.

[5] H. A. P. Blom and Y. Bar-Shalom
"The interacting multiple model algorithm for systems with Markovian switching coefficients,"
IEEE Trans. Autom. Control, vol. 33, no. 8, pp. 780–783, Aug. 1988.

[6] A. Bole, A. Wall, and A. Norris
"Chapter 5–Automatic identification system (AIS),"
in *Radar and ARPA Manual*, A. Bole, A. Wall, and A. Norris, Eds., 3rd ed. Oxford, U.K.: Butterworth-Heinemann, 2014, pp. 255–275.

[7] E. F. Brekke, A. G. Hem, and L.-C. N. Tokle
"Multitarget tracking with multiple models and visibility: Derivation and verification on maritime radar data,"
IEEE J. Ocean. Eng., vol. 46, no. 4, pp. 1272–1287, Oct. 2021.

[8] E. F. Brekke, E. F. Wilthil, B.-O. H. Eriksen, D. K. M. Kufoalor, Ø. K. Helgesen, I. B. Hagen, M. Breivik, and T. A. Johansen
"The Autosea project: Developing closed-loop target tracking and collision avoidance systems,"
J. Phys.: Conf. Ser., vol. 1357, pp. 1–12 2019.

[9] C. Carthel, S. Coraluppi, and P. Grignan
"Multisensor tracking and fusion for maritime

surveillance,”
in *Proc. 10th Int. Conf. Inf. Fusion*, 2007, pp. 1–6.

[10] K. Cutlip
Spoofing: One Identity Shared by Multiple Vessels. Global fishing watch, 2016. https://globalfishingwatch.org/data/spoofing-one-identity-shared-by-multiple-vessels/.

[11] J. Dezert and Y. Bar-Shalom
“Joint probabilistic data association for autonomous navigation,”
IEEE Trans. Aerosp. Electron. Syst., vol. 29, no. 4, pp. 1275–1286, Oct. 1993.

[12] T. E. Fortmann, Y. Bar-Shalom, and M. Scheffe
“Sonar tracking of multiple targets using joint probabilistic data association,”
IEEE J. Ocean Eng., vol. 8, no. 3, pp. 173–184, Jul. 1983.

[13] D. Gaglione, P. Braca, G. Soldi, F. Meyer, F. Hlawatsch, and M. Z. Win
“Fusion of sensor measurements and target-provided information in multitarget tracking,”
IEEE Trans. Signal Process., vol. 70, pp. 322–336, Dec. 2022.

[14] I. R. Goodman, H. T. Nguyen, and R. Mahler
Mathematics of Data Fusion. Dordrecht, NL, USA: Springer, 1997.

[15] K. Granström, L. Svensson, Y. Xia, J. Williams, and Á. F. García-Femández
“Poisson multi-Bernoulli mixture trackers: Continuity through random finite sets of trajectories,”
in *Proc. 21st Int. Conf. Inf. Fusion*, 2018, pp. 1–5.

[16] G. Grimmett and D. Stirzaker
Probability and Random Processes. Oxford, U.K.: Oxford University Press, 2001.

[17] B. Habtemariam, R. Tharmarasa, M. McDonald, and T. Kirubarajan
“Measurement level AIS/radar fusion,”
Signal Process., vol. 106, pp. 348–57, 2015.

[18] A. G. Hem and E. F. Brekke
“Variations of joint integrated data association with radar and target-provided measurements,” [source code]. Available: https://doi.org/10.24433/CO.4125751.v1.

[19] Radiocommunication Sector of ITU, “Technical characteristics for an automatic identification system using time division multiple access in the VHF maritime mobile frequency band,” ITU-R M.1371-5, Recommendation ITU-R M.1371-5, Feb. 2014.

[20] X. R. Li and Y. Bar-Shalom
“Design of an interacting multiple model algorithm for air traffic control tracking,”
IEEE Trans. Control Syst. Technol., vol. 1, no. 3, pp. 186–194, Sep. 1993.

[21] E. Liland
“AIS aided multi hypothesis tracker—multi-frame multi-target tracking using radar and the automatic identification system,” Master’s thesis, NTNU, Trondheim, Norway, 2018.

[22] M. Longbin, S. Xiaoquan, Z. Yiyu, S. Z. Kang, and Y. Bar-Shalom
“Unbiased converted measurements for tracking,”
IEEE Trans. Aerosp. Electron. Syst., vol. 34, no. 3, pp. 1023–1027, Jul. 1998.

[23] F. Meyer, P. Braca, P. Willett, and F. Hlawatsch
“A scalable algorithm for tracking an unknown number of targets using multiple sensors,”
IEEE Trans. Signal Process., vol. 65, no. 13, pp. 3478–3493, Jul. 2017.

[24] D. Musicki and R. Evans
“Joint integrated probabilistic data association: Jipda,”
IEEE Trans. Aerosp. Electron. Syst., vol. 40, no. 3, pp. 1093–1099, Jul. 2004.

[25] D. Musicki and S. Suvorova
“Tracking in clutter using IMM-IPDA-based algorithms,”
IEEE Trans. Aerosp. Electron. Syst., vol. 44, no. 1, pp. 111–126, Jan. 2008.

[26] P. C. Niedfeldt, K. Ingersoll, and R. W. Beard
“Comparison and analysis of recursive-RANSAC for multiple target tracking,”
IEEE Trans. Aerosp. Electron. Syst., vol. 53, no. 1, pp. 461–476, Feb. 2017.

[27] J. Å. Sagild, A. G. Hem, and E. F. Brekke
Counting technique versus single-time test for track-to-track association,
in *Proc. 24th Int. Conf. Inf. Fusion*, 2021, pp. 1–7.

[28] D. Schuhmacher, B. T. Vo, and B. N. Vo
“A consistent metric for performance evaluation of multi-object filters,”
IEEE Trans. Signal Process., vol. 56, no. 8, pp. 3447–3457, Aug. 2008.

[29] M. Schuster, M. Blaich, and J. Reuter
“Collision avoidance for vessels using a low-cost radar sensor,”
IFAC Proc. Volumes, vol. 47, no. 3, pp. 9673–9678, 2014.

[30] R. L. Streit
Poisson Point Processes: Imaging, Tracking, and Sensing. New York, NY, USA: Springer, 2010.

[31] T. Tengesdal, L. M. Millefiori, P. Braca, and E. F. Brekke
“Joint stochastic prediction of vessel kinematics and destination based on a maritime traffic graph,”
in *Proc. Int. Conf. Elect., Comput., Commun. Mechatronics Eng.*, 2022, pp. 1–8.

[32] B.-T. Vo and B.-N. Vo
“Labeled random finite sets and multi-object conjugate priors,”
IEEE Trans. Signal Process., vol. 61, no. 13, pp. 3460–3475, Jul. 2013.

[33] X. Wang, S. Challa, and R. Evans
“Gating techniques for maneuvering target tracking in clutter,”
IEEE Trans. Aerosp. Electron. Syst., vol. 38, no. pp. 1087–1097, Jul. 2002.

[34] J. L. Williams
“Marginal multi-Bernoulli filters: RFS derivation of MHT, JIPDA, and association-based MeMBer,”
IEEE Trans. Aerosp. Electron. Syst., vol. 51, no. 3, pp. 1664–1687, Jul. 2015.

[35] E. F. Wilthil, Y. Bar-Shalom, P. Willett, and E. F. Brekke
“Estimation of target detectability for maritime target tracking in the PDA framework,”
in *Proc. 22nd Int. Conf. Inf. Fusion*2019, pp. 1–8.

[36] E. F. Wilthil, E. F. Brekke, and O. B. Asplin
“Track initiation for maritime radar tracking with and without prior information,”
in *Proc. 21st Int. Conf. Inf. Fusion*, 2018, pp. 1–8.

[37] E. F. Wilthil, A. L. Flåten, and E. F. Brekke
“A target tracking system for ASV collision avoidance based on the PDAF,”
Sensing and Control for Autonomous Vehicles. Berlin, Germany: Springer, 2017, pp. 269–288.

Audun G. Hem received his M.Sc. degree in engineering cybernetics from the Norwegian University of Science and Technology (NTNU), Trondheim, Norway, in 2021, and he is currently a Ph.D. student at the Department of Engineering Cybernetics, NTNU. He specializes in target tracking for autonomous surface vehicles and is currently working on the Autosit project.

Edmund F. Brekke received the M.Sc. degree in industrial mathematics and the Ph.D. degree in engineering cybernetics from the Norwegian University of Science and Technology (NTNU), Trondheim, Norway, in 2005 and 2010, respectively. From 2010 to 2014, he worked with the Acoustic Research Laboratory (ARL), NUS, Singapore, as a postdoctoral Research Fellow. In 2014, he rejoined NTNU and the Department of Engineering Cybernetics as an Associate Professor in sensor fusion. This position was funded as a gift professorship by DNV in 2014–2019. He was the Project Manager for the Autosea project, developing methods for sensor fusion and collision avoidance for autonomous surface vehicles. He is currently a Key Scientist for the Autoferry project, which is focused on autonomous pedestrian ferries, Work Package Manager in the Centre for Research-Based Innovation SFI Autoship, and the Project Manager of the Autosit and Autosight projects, which are focused on automated situational awareness for autonomous surface vehicles. He is an Associate Editor of the IEEE Journal of Oceanic Engineering. His research interests are in the area of sensor fusion and situational awareness, with a particular focus on multitarget tracking.

Probabilistic Vehicle Tracking with Sparse Radar Detection Measurements

PHILIPP BERTHOLD
MARTIN MICHAELIS
THORSTEN LUETTEL
DANIEL MEISSNER
HANS-JOACHIM WUENSCHE

Most automotive perception systems leverage radar sensors for their long-range measuring capability and weather robustness at economic costs. A downside is the rather low spatial resolution. It complicates the estimation of pose and size of an extended object. High-resolution sensors facilitate techniques like shape recognition based on a single measurement. But even these sensors only provide sparse measurements at larger distances, which makes instantaneous object detection highly ambiguous. We propose an approach that incorporates the current state estimate to probabilistically identify the true origin of a detection and thereby decreases its association ambiguity. It uses all given measurement data, including the radial speed. This improves the information gain for mass-market sensors with a high measurement uncertainty. We first perform a parametrization of the object using a set of components. They describe the characteristics of a detection in dependency of the current state estimate and various physical relations. Their superposition resembles the spatial detection likelihood of the entire object. Subsequently, we perform a computationally efficient state update that exploits the probabilistic association of the detection to the components. All steps take about 20 μs of computing time. In this article, we demonstrate this technique in an application that tracks vehicles with radar detections. Besides providing details on the algorithm and a formal description of the components, we also illustrate the probabilistic association with examples. Finally, we discuss the performance in real-world tracking scenarios and outline interfaces to multi-hypotheses and multi-sensor fusion algorithms. This paper is accompanied by an exemplary MATLAB implementation and a demonstration video.

Manuscript received February 17, 2022; revised September 27, 2022; released for publication March 20, 2023.

P. Berthold, M. Michaelis, T. Luettel, and H.-J. Wuensche are with the Institute for Autonomous Systems Technology (TAS), University of the Bundeswehr Munich, 85577 Neubiberg, Germany (e-mail: philipp.berthold@unibw.de; martin.michaelis@unibw.de; thorsten.luettel@unibw.de; joe.wuensche@unibw.de).

D. Meissner is with Objects, Obstacles and AI based low level Perception, BMW Group, 80788 Munich, Germany (e-mail: daniel.da.meissner@bmw.de).

1557-6418/22/$17.00 © 2022 JAIF

I. INTRODUCTION

Advanced automotive perception systems have to meet high expectations in terms of cost-effectiveness, performance, and robustness. The fusion of different sensor types accommodates these requirements by exploiting the cumulative strengths. Monocular cameras are widely used to identify objects as they provide semantic information. However, they do not provide range measurements. This impairs the immediate estimation of the position and the extent of objects and often requires the incorporation of model knowledge. Stereo cameras are typically limited to short ranges [11]. On the other hand, LiDAR sensors mainly capture high-resolution spatial information on an object's contour, which encapsulates the pose and the extent of an object. Radar sensors also provide full spatial information, but with a lower resolution. However, their major advantages are their ability to directly measure radial speed and their resistance to tough weather conditions due to their lower frequency range.

The nature of *extended objects* states that multiple detections might be caused by arbitrary parts of the extent of the object. High-resolution sensors provide such a large quantity of detections that the contour of objects can be spotted in a single measurement [8]. The thereby captured object instances can be directly filtered to their corresponding tracks [28]. However, these approaches are not feasible if only sparse measurement data are available, resulting in few or no resolved detections per object. This issue is not necessarily limited to mass-market sensors; also high-performance sensors only provide sparse measurement data at respective distances. At this point, a contour (or structure) extraction from a single measurement is no longer possible. To sustain the tracking, the detections need to be directly filtered to their tracks. The arising challenge is the correct determination of the origin of each detection without any structural information from the current measurement data. Especially in the case of a radar sensor, the association problem is tough: The lateral measurement noise is substantial due to its measurement principle [25] and depends on the complexity of the surrounding.

We propose a filtering approach that tackles this association problem. First, it splits the object in components with individual, physically deduced detection characteristics. Second, it incorporates the current object state estimate to model the current statistical appearance of these components. We apply this approach to radar sensors for vehicle tracking in this work. The utilized radar sensors provide a set of points, which is called *scatter* data. Each point represents a so-called *detection*, which represents a maximum of local reflectivity and is given by measurements of position, radial speed, and amplitude. Our approach not only uses the position measurement but also exploits the radial speed measurement. This shifts the association problem to a space of higher dimension and improves its resolution.

A. Related Work

Adaptive cruise control has been one of the first popular automotive radar applications. The radar detects vehicles in direction of travel and returns their distance and speed. For this application, it is sufficient to obtain a single measurement for an object. Modern advanced driver assistance systems like the lane change assistant require pose and extent information. Filter techniques, which use radar measurements to estimate both pose and extent of objects, can be grouped into four categories. In 2017, Granström *et al.* [14] defined three categories: generic spatial models, set of points on a rigid body (SPRB) models, and physical models. In the past few years, however, machine learning approaches have also been adopted in this area and represent a popular fourth category.

Spatial models define the extent by a shape or function. Popular approaches incorporate ellipse-based models [24], probabilistic density fields given by Poisson processes [12], [13], radius functions [14], [38], or any kind of functional shape descriptions by Gaussian processes [27], [35]. As these approaches can be expressed in closed functions, they show high runtime performance. Additionally, they do not require explicit model knowledge and are therefore suitable for a wide spectrum of objects. However, the lack of model knowledge impairs the extent estimation accuracy and, consequently, also the estimation of the pose and the kinematics. The impact of the observation perspective and model-specific features like micro-Doppler cannot be exploited.

SPRB models use discretized spatial model descriptions instead of continuous ones. According to SPRB, the object can be modeled by a set of discrete scattering points. The location of these points can be estimated online [16], or by incorporating some model knowledge. In this manner, Bühren *et al.* [9] place the points on typical reflection sources of vehicles like wheels and corners. They also consider some visibility constraints. Hammarstrand *et al.* [17] propose an adequate SPRB filtering approach. A major downside of SPRB is the missing ability to model continuous, extended parts of the object.

Physical models are powerful and accurate in describing the object and predicting its expected measurements. They are often composed of an object model and a measurement model. The accuracy of the object model varies from geometric shapes to 3D computer models. The measurement model is an inference of physical considerations. Ray tracing methods [23] incorporate any desired level of model knowledge and achieve high reproduction accuracy. The prevalent downside is a substantial runtime overhead, which often renders them futile in real-time multi-object tracking applications. The poor runtime performance is not only due to the demanding modeling computation but also due to the tracking itself that often requires particle filters.

A both new and by now very popular approach to model radar detections is machine learning. It correlates the state of objects to their obtained detection characteristics in annotated training data. These approaches vary from variational Gaussian mixtures (VGM) [19], [21], [30], [37] to deep neural networks [10], [36]. The latter was facilitated by the recent progress in 4D high-resolution imaging radars that provide a large number of detections per target in a single measurement frame [41]. Machine learning approaches allow accurate measurement reproductions and circumvent expensive manual statistical studies on the sensor model. The sensor-specific measurement characteristics are learned from the measurement data. Their overall performance depends on the spectrum of the scenarios in the training data. If the training data does not contain more complex scenarios like different kinds of occlusion, then the outcome is undefined. Additionally, the network needs to learn new training data to adapt to new sensors or object types; it cannot be parametrized easily. However, thanks to recent advances in GPU development, their runtime performance allows real-time usage.

B. Previous Research

With the exception of some machine learning approaches, most of these models do not really match our observed data. This seems to be mainly due to oversimplification or incomplete modeling of the objects, i.e., vehicles. As a result, our aim is the development of a model that is physically derived to ensure generalizability. Its abstraction is chosen at a level that allows for its real-time usage in tracking applications, but without sacrificing performance potential.

Our work has began with a radar measurement analysis. In [2], we performed measurement campaigns to record the reflection characteristics of vehicles. These campaigns cover a spectrum of relative poses between the radar sensor and the target vehicle. An algorithm, which sorts, accumulates, and statistically re-weighs the measurement data, extracts a detection probability map in target coordinates for any desired relative observation pose. These results reveal a high impact of the observation angle. Unsurprisingly, the outer parts of the vehicle, which are oriented perpendicular toward the radar sensor, cause the most significant portion of the object's reflectivity. Moreover, the corners are highly reflective as a part of the round curve is always perfectly orthogonal toward the sensor. Next to the vehicle sides and corners, the wheels are also significant reflection sources. Wheels that are facing toward the radar reflect well due to the wheel rim. But the measurement analysis reveals that the opposite wheels are also often spotted in the radar measurement data. The low mounting height of series radars often causes a line of sight between the sensor and the opposite wheels. Opposite-wheel visibility is also given by underbody reflections, i.e., depending on the elevation angle of incidence, the beam is reflected by the ground surface and the vehicle underbody. This effect causes a slight spatial detection probability for the complete underbody extent of the vehicle. On the contrary, inner parts of the vehicle are rarely visible. Varnish and windows heavily attenuate the beam amplitude. Roof structures are visible if the vertical field of view of the sensor is sufficiently large.

In [3], we complemented this work with a radar measurement model. Its primary aim is the preferably accurate probabilistic prediction of measurement data for any given target state. The measurement model is a physical one. It achieves a generic measurement reproduction and an inherent incorporation of effects like mutual occlusion. Typically, the latter is hard to accomplish when dealing with statistical or oversimplified models. Our model separates the generation process of detection measurements in abstraction layers like physical wave distribution, signal reception, and peak detection. We utilize this measurement model to evaluate models against real-world data, but its runtime performance hinders an immediate usage in multi-hypotheses tracking applications.

In [4], we enhanced both the measurement analysis and the measurement model with the Doppler-derived radial speed measurement. The radial speed measurement provides valuable information as it is directly measured and subject to only low measurement noise. Besides, it plays a crucial role in determining the angle of the detections. The radial speed measurement can be predicted for any point of the rigid object body as long as the relative kinematics of the object and the radar sensor are known. Parts like legs or wheels that move relative to the rigid body span a range of potential Doppler measurements. The radial speed measurements of moving parts of a moving object are known as micro-Doppler measurements and are subject to ongoing research [18], [32]. Current approaches [20] explicitly detect micro-Doppler measurements of vehicles in imaging radar data, extract the wheels by exploiting the Doppler spectrum [39], and use their position to track their pose.

C. Our Contribution

While our previous work has primarily elaborated a preferably precise and well-founded but computationally expensive physical model, this article presents its abstraction that can be utilized in real-time tracking applications. As far as possible, its functional structure is derived from physical and technical interrelations. According to our findings, the division of an object into different components yields a good modularizability and allows for individual measurement characteristics. The proposed abstracted model uses this mechanism and therefore resembles primarily the SPRB approaches. The main differences are that our model also supports and utilizes spatially extended, continuous components. Besides, we exploit the kinematic measurement of a

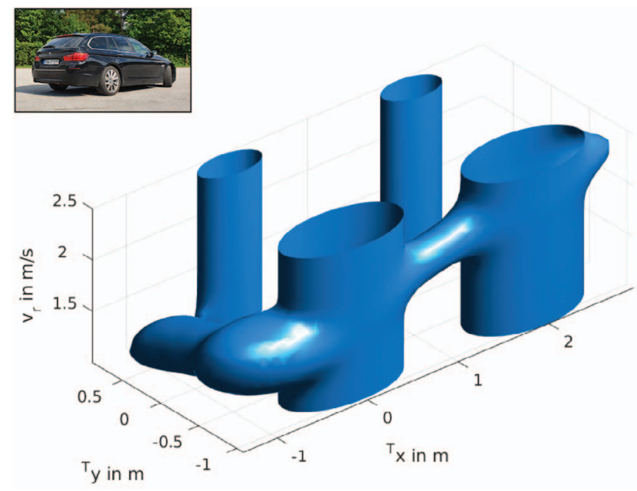


Fig. 1. The expected spatial detection likelihood of a moving vehicle in a left turn. The measurement space consists of the two-dimensional Euclidean position in target coordinates $^{T}(x, y)$ and the Doppler-deduced radial speed measurement v_r shown in the z-axis. To illustrate the three-dimensional detection likelihood, this plot shows the isosurface of an examplary detection likelihood. The vertically extended tubes are caused by the micro-Doppler effect of the wheels. The azimuth view angle corresponds to the observation angle of the radar sensor. Due to the left turn, the front of the vehicle moves away faster than the rear.

detection, e.g., the radial speed, to improve the origin search and to perform a direct kinematic state update. Figure 1 outlines the modeled joint measurement space and illustrates the expected multidimensional detection likelihood of a moving vehicle. The state of the components is linked to the state vector using type-dependent definitions. Concerning the components themselves, we consider not only angular visibility regions (similar as proposed by [9]) but also the reflectivity, the kinematic Doppler properties, and physical effects like scattering. This addition of features, though, requires more modeling effort. The proposed spatial measurement function described in this article can be used independently from the proposed tracking approach in any Bayes-based filters. It also outputs the expected number of detections for any given object state.

Learning-based approaches have the essential advantage that learning the reflectivity of an object does not require expert knowledge. However, learning methods usually learn the complete stack of measurement generation and cannot split different components, e.g., sensor model from object model. The proposed approach aims for parameterability and exchangeability of all relevant modules. New object types can be supported by partial adjustments of the object model on the basis of a datasheet, for example. There is no need for gathering and annotating new training data and relearning, especially if only partial properties have to be adjusted. Physical effects like scattering are mathematically described and therefore generically utilizable, and it is possible to apply assumptions like symmetry for a subset of the object model. Another drawback is the computing effort that comes, i.e., with the high dimensionality of the VGM. Scheel *et al.* [30] still manage, though, to incorporate the radial speed profile of an object. The origin association and object tracking is commonly performed using probabilistic multi-hypothesis tracking (PMHT) [33], particle filters or labeled multi-Bernoulli filter (LMB) [30]. In case of PMHT applications, the computation of the association probabilities [7, eq. (19)] is similar to our approach. The subsequent tracking, however, optimizes the association cost using the expectation-maximization algorithm. We use a probabilistic Kalman-based approach [5]: The usage of explicit mathematical functions allows for a very fast execution time of our approach. The complete state update takes about 20 µs, rendering it suitable for multi-hypotheses tracking applications. Besides, our approach inherently supports and utilizes extended structures of the object model like vehicle sides. This avoids their approximation with a large number of Gaussian mixture components and reduces both computing effort and bias effects.

Another key functionality of the proposed filter is its treatment of sparse measurement data. In contrast to applications requiring imaging or high-resolution data, where, e.g., wheels can be spotted in a single measurement, this filter is suitable for radars receiving about two detections per target and measurement epoch on average. It performs a probabilistic association and uses the state estimates to infer the origin of the detections.

We provide an implementation of the proposed filtering approach for vehicle tracking and finally a discussion of its tracking performance based on real-data examinations. We also provide a MATLAB code that implements the proposed object modeling and state update.[1]

D. Structure

The goal of the presented approach is to update the state $\mathbf{x}$ using radar detection measurements $\mathbf{y}$ at timestamp k:

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) \propto p(\mathbf{y}_k|\mathbf{x}_k) \cdot p(\mathbf{x}_k|\mathbf{y}_{1:k-1}). \tag{1}$$

To solve the origin search problem, we model the target object as a complex of spatially distributed scattering sources (*components*). They show different statistical properties in terms of detection rates and kinematic measurements. We exploit this heterogeneity to obtain statistical inference regarding the possible origin of the detection. The division of the object into these components $j \in J$ resembles Gaussian mixtures, allowing the marginal measurement likelihood to be specified in the following format, where $\mathrm{o}^{(j)}$ is the mixture weight:

$$p(\mathbf{y}_k|\mathbf{x}_k) \propto \sum_{j\in J} \mathrm{o}^{(j)}(\mathbf{x}) \cdot \mathcal{N}(\cdot)\,. \tag{2}$$

[1] Available at https://github.com/UniBwTAS/sparse_radar_tracking.

Table I
The Notation and Some Variables of This Article

Symbol	Description
$\mathbf{x}$	(target) state vector
$\mathbf{y}$	single detection measurement
$\mathbf{x}^\star$	predicted state (prior)
$\mathbf{y}^\star$	predicted measurement (given $\mathbf{x}^\star$)
$\hat{\mathbf{x}}$	updated state (posterior)
x, y	scalar Euclidean coordinates
(x, y)	Euclidean position vector (2D)
${}^{F}(\cdot)$	reference coordinate frame $F \in \{$ world frame W, ego vehicle frame E, sensor frame S, target object frame T$\}$
$(\cdot)^{(K)}$	component identifier K

Each component thereby abstracts technical principles that strongly depend on the object type and its current state.

This article is structured bottom-up: Section II specifies the notation and the utilized variables. Section III states our sensor model. Section IV denotes the object modeling concept and its implementation for a vehicle. The fundamental technical properties for each component are stated. Section V carries this on to a spatial measurement function that resembles the marginal measurement likelihood. It is given for each component; the superposition of all likelihoods describes the spatial measurement function for the complete object. This measurement function can be used in Bayes filters. In Section VI, we utilize the predictive measurement likelihood $p(\mathbf{y}_k|\mathbf{y}_{1:k-1})$ based on the state prediction to determine the origin for a given detection measurement $\mathbf{y}$ in a probabilistic manner. Section VII proposes our Bayesian filtering approach. It explains the state update of a single state hypothesis. The performance of the algorithm is then examined with real-world tracking scenarios in Section VIII. We briefly outline interfaces to multi-object multi-sensor tracking frameworks in Section IX and finally discuss the filter in Section X. This article is concluded in Section XI and provides an outlook for future work in Section XII.

II. NOTATION, VARIABLES, AND COORDINATE FRAMES

Table I briefly outlines frequently used variables and the notation of this article. The therein referenced coordinate frames are required for coordinate transformations of the detection measurements and illustrated in Fig. 2. A detection measurement $\mathbf{y}$ is initially obtained in polar sensor coordinates and consists of

- range measurement r with measurement noise σ_r,
- angle measurement α with measurement noise σ_α,
- Doppler measurement $\dot{r}$ with measurement noise $\sigma_{\dot{r}}$,
- amplitude or radar cross section measurement a.

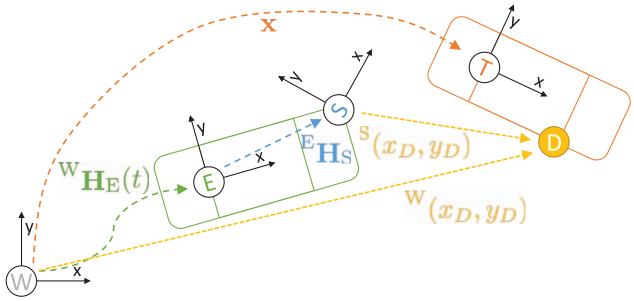


Fig. 2. The coordinate systems. The odometry provides a time-variant transform ${}^{\mathrm{W}}\mathbf{H}_{\mathrm{E}}(t)$ from the world coordinates W to the ego vehicle coordinates E. The static mounting location ${}^{\mathrm{E}}\mathbf{H}_{\mathrm{S}}$ of the treated sensor establishes the sensor coordinate system S. These transforms project the detection measurement D given in sensor coordinates ${}^{\mathrm{S}}(x_D, y_D)$ into world coordinates ${}^{\mathrm{W}}(x_D, y_D)$. The state estimate $\mathbf{x}$ provides the reference point of the target T in world coordinates.

The position of the detection in Cartesian sensor coordinates ${}^{\mathrm{S}}(x_D, y_D)$ is given by

$$ {}^{\mathrm{S}}(x_D, y_D) := {}^{\mathrm{S}}\begin{bmatrix} x_D \\ y_D \end{bmatrix} = {}^{\mathrm{S}}\begin{bmatrix} r \cdot \cos\alpha \\ r \cdot \sin\alpha \end{bmatrix}, \tag{3} $$

and the Cartesian measurement noise matrix in sensor coordinates ${}^{\mathrm{S}}\mathbf{R}_{\mathbf{xy}}$ is

$$ {}^{\mathrm{S}}\mathbf{R}_{\mathbf{xy}} = \mathrm{R}(\alpha) \cdot \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & (2r \cdot \tan(\sigma_\alpha/2))^2 \end{bmatrix} \cdot \mathrm{R}(\alpha)^{\mathsf{T}}, \tag{4} $$

where $\mathrm{R}(\cdot)$ is the two-dimensional rotation matrix. These conversions are subject to bias effects when used in state filters. Dedicated compensation techniques are provided by Bordonaro *et al.* [6], but these effects are negligible compared to real-world measurement phenomena the filter has to deal with in this application. A transformation matrix ${}^{\mathrm{E}}\mathbf{H}_{\mathrm{S}}$ from sensor coordinates S to ego coordinates E, which reflects the mounting position, and a time-variant egomotion transformation matrix ${}^{\mathrm{W}}\mathbf{H}_{\mathrm{E}}(t)$ from ego to world coordinates convert these parameters into world coordinates. The position of the detection in world coordinates ${}^{\mathrm{W}}(x_D, y_D)$ is given by

$$ {}^{\mathrm{W}}\begin{bmatrix} x_D \\ y_D \end{bmatrix} = {}^{\mathrm{W}}\mathbf{H}_{\mathrm{E}}(t) \cdot {}^{\mathrm{E}}\mathbf{H}_{\mathrm{S}} \cdot {}^{\mathrm{S}}\begin{bmatrix} x_D \\ y_D \end{bmatrix}. \tag{5} $$

Therefore, its measurement noise in world coordinates can be derived as

$$ {}^{\mathrm{W}}\mathbf{R}_{\mathbf{xy}} = \mathrm{R}\left({}^{\mathrm{W}}\varphi_{\mathrm{E}}(\mathrm{t}) + {}^{\mathrm{E}}\varphi_{\mathrm{S}}\right) \cdot {}^{\mathrm{S}}\mathbf{R}_{\mathbf{xy}} \cdot \mathrm{R}\left({}^{\mathrm{W}}\varphi_{\mathrm{E}}(\mathrm{t}) + {}^{\mathrm{E}}\varphi_{\mathrm{S}}\right)^{\mathsf{T}}, \tag{6} $$

with the mounting yaw ${}^{\mathrm{E}}\varphi_{\mathrm{S}}$ being deduced from ${}^{\mathrm{E}}\mathbf{H}_{\mathrm{S}}$ and the heading of the ego vehicle ${}^{\mathrm{W}}\varphi_{\mathrm{E}}(t)$ provided by ${}^{\mathrm{W}}\mathbf{H}_{\mathrm{E}}(t)$.

Further variables in this article, like the state vectors, are explained when they are introduced.

III. SENSOR MODEL

This section briefly denotes some sensor-specific parameters and their derivations. Given a specific target reflector, the radar sensor will measure a detection with a particular detection probability. It manifests in the *detection rate* o, which describes at which rate a detection is invoked by a specific reflectivity at a certain distance r. As this rate depends both on the sensor and the individually measured reflectivity, we factor out the sensor-specific part: the *reference rate* $\mathrm{o_R}(r)$. We address not only the resolution ability of the sensor but also firmware-sided tuning. In fact, the signal strength of a radar echo decreases with the fourth power of the distance, but the firmware often neutralizes this effect by applying adapted trigger and noise thresholds in the constant false alarm rate algorithms [29]. This way, the detection rate of an object keeps almost constant over the distance, until the maximum measurement range of the sensor is reached. At this point, the detection rate drops rapidly. However, the exact effect should be determined by measurement analysis for each sensor model. We model the reference rate $\mathrm{o_R}(r)$ of our sensor with

$$ \mathrm{o_R}(r) = a \cdot \mathrm{erf}\left((r_{\max} - r)/d\right), \tag{7} $$

using the error function $\mathrm{erf}(\cdot)$

$$ \mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x \mathrm{e}^{-t^2} \mathrm{d}t, \tag{8} $$

and the parameters effective maximum range $r_{\max}$ (e.g., 40 m), decay magnitude d (e.g., 10 m) and amplitude a, which depends on the reference reflectivity. The reference reflectivity can be chosen arbitrarily here; we have selected the corner of a car. If the amplitude a is determined by measurement analysis, then the reference rate already incorporates the ratio for false negative and true positive detections. The false positive detection rate causes clutter measurements and needs to be determined independently.

The detection measurement is spread around the true position of the target according to its measurement noise characteristics. They are mostly given in sensor coordinates to match the physical measurement process and are described by the scalar uncertainties introduced in Section II. Some sensors provide these uncertainties for each measured detection themselves. This allows for the incorporation of certain ambiguities in the sensor-internal preprocessing that depend on the environment [4].

Another important parameter is the bandwidth of the sensor. It affects the resolution capability. Besides the number of resolved detections, it also determines a kind of longitudinal “penetration” depth: The larger the range resolution is, the more reflections of a larger depth of the object (e.g., caused by vertical tapers) are received inseparably. The peak detection of the sensor then signals a longitudinal measurement, which is smeared over the penetration depth, and therefore needs to be considered as an additive term of the range measurement. The actual incorporation of this term is firmware-specific and depends on the structure of the target.

These parameters describe the radar sensor and aim for its exchangeability without the need to touch the object model. However, if a sensor is utilized that changes the relative detection rate ratio between the individual components of the object, an adjustment of the object model might be required, since this ratio is primarily exploited in the presented approach. We have not observed such a behavior in comparable sensor series, though. Obtaining the extrinsic and intrinsic sensor parameters by dedicated measurement campaigns and ground-truth-assisted analysis is often not trivial despite automation, as it requires manual effect decomposition and can take several days.

IV. OBJECT MODEL AND A VEHICLE IMPLEMENTATION

The abstraction of the physical model (i.e., [3], [4]) has to meet diverse requirements. Its usage in a multi-hypotheses tracking framework must meet real-time constraints, although an oversimplification of the precise physical model impairs the tracking performance. A major criterion is the correct reproduction of the object contour, since any deviation causes a position bias of the object estimate. The measurement data show a strong impact of the viewing angle, which thus needs to be modeled accordingly. Additionally, the data does not only show a dependency on occlusions by other objects but also on self-occlusions. For instance, wheels that are facing the sensor might shadow opposite wheels.

Our approach is to model an object by the superposition of a certain number of *components*: the object is split into a set of separate, individually, and formally described parts. This method gives the opportunity to use different measurement models, visibility constraints, and detection likelihoods for each component. The number of components should preferably be small to achieve fast computation times, but sufficiently large to allow a precise representation of the object. In the example of our vehicle model, suitable component classes are wheels, corners, sides, and the body. A component class can have multiple instances. Physical effects like the micro-Doppler can be specifically implemented for each class and can be exploited to tightly associate a detection to a component. The visibility constraint of a component class can also depend on other components in advanced models. This enables the modeling of self-occlusions or, in multi-object tracking applications, occlusions by other objects.

The following describes a set of components that jointly define the measurement characteristic of a vehicle. This set and its configuration have been obtained by recording and analyzing the reflection characteristics of various vehicles, among them a compact class vehicle and a sport utility vehicle in particular as edge cases [2]. Besides, short-range and far-range sensors have been utilized. According to our findings, the set of components comprising wheels, corners, sides, and the body is a good compromise between precision and complexity regarding the utilized radar sensors. This set results in 4 component classes and 13 component instances. Each component class is defined by a set of attributes $A^{(\cdot)} = \{$ position ${}^{\mathrm{T}}(x_C, y_C)^{(\cdot)}(\mathbf{x})$, position uncertainty $\mathrm{Cov}\left({}^{\mathrm{T}}(x_C, y_C)^{(\cdot)}\right)$, detection rate $\mathrm{o}^{(\cdot)}(\mathbf{x})$, radial speed model $v_r^{(\cdot)}(\mathbf{x})\}$. The *position* ${}^{\mathrm{T}}(x_C, y_C)^{(\cdot)}(\mathbf{x})$ denotes the position of the component in the target frame T. The *position uncertainty* $\mathrm{Cov}\left({}^{\mathrm{T}}(x_C, y_C)^{(\cdot)}\right)$ describes the uncertainty of this position and can also be used to model a slight extent with an additive noise term. The detection rate $\mathrm{o}^{(\cdot)}(\mathbf{x})$ describes the expected number of detections this component invokes. It depends mainly on the reflection characteristics of the component and its visible angular extent. The visible angular extent is usually estimated using the target state and the pose of the radar sensor toward the object. The radial speed model $v_r^{(\cdot)}(\mathbf{x})$ denotes the measurement model of the radial speed measurements for the given component. We deduce the mathematical correlation of these attributes to the state vector using the physical relations found in [3], and parametrize those accordingly to match the observed data. In this work, we use the plain extent state vector $\mathbf{x}_{\mathrm{ext}} = [l, w]^\intercal$ to estimate the length l and the width w of the object to preserve a low computing expense. As a result, all remaining required information, like the wheel positions, is statistically derived from both variables. Alternatively, any desired parameter can also be included in the state vector. Apart from that, the following component descriptions are based on a constant turnrate and velocity (CTRV) state model, which describes the position (x, y) and heading φ of the object along with its kinematical properties translational speed v and yaw rate ω. The state vector is finally given as

$$\mathbf{x}_{\mathrm{kin}} = [x, y, \varphi, v, \omega]^\intercal, \tag{9}$$

$$\mathbf{x} = [\mathbf{x}_{\mathrm{kin}}^\intercal, \mathbf{x}_{\mathrm{ext}}^\intercal]^\intercal. \tag{10}$$

The kinematic transition matrix yields [31], [34]:

$$\mathbf{x}_{\mathrm{kin},k+1} = \begin{bmatrix} x_k + v_k/\omega_k \cdot (+\sin(\omega_k \Delta t + \varphi_k) - \sin(\varphi_k)) \\ y_k + v_k/\omega_k \cdot (-\cos(\omega_k \Delta t + \varphi_k) + \cos(\varphi_k)) \\ \varphi_k + \omega_k \Delta t \\ v_k \\ \omega_k \end{bmatrix}, \tag{11}$$

and predicts the state epoch $k+1$ from epoch k by integrating the sample time Δt. When the yaw rate ω is close to zero, the transition matrix should be simplified to avoid numeric issues:

$$\mathbf{x}_{\mathrm{kin},k+1} = \begin{bmatrix} x_k + v_k \cdot \cos(\varphi_k)\Delta t \\ y_k + v_k \cdot \sin(\varphi_k)\Delta t \\ \omega_k \Delta t + \varphi_k \\ v_k \\ \omega_k \end{bmatrix}. \tag{12}$$

The transition matrix of the extent state vector $\mathbf{x}_{\mathrm{ext}}$ for rigid objects is the unit matrix $\mathbf{1}_{2\times 2}$.

In the following, the attributes for the four component classes—corners, wheels, sides, and body—are declared.

A. Vehicle Corners

The corners of a vehicle benefit from a good reflection effect. As discussed in Section I-B, the curvature of a visible corner exposes a spot that is perfectly perpendicular to the line of sight of the radar sensor. This spot reflects electromagnetic waves back to the radar sensor with only minimal deflection and thereby obtains an excellent visibility in the measurement data. Consequently, we model the corner as a point target. The exact position of the reflective spot depends on the shape of the vehicle and is empirically derived from the extent state vector. We model the uncertainty of the position with an additive Gaussian noise term. The resulting description is noted in Table II and graphically represented in Fig. 3.

The detection rate depends on several factors, where each models a specific influence on the detection rate. The corner detection rate $\mathrm{o}^{(C)}$ depends here on three factors. The first one is the reference rate, $\mathrm{o}_{\mathrm{R}}(r)$, as discussed in Section III. The second factor, the *component base rate* o_{C}, now puts the reflectivity of a component into relation to the reference reflectivity. The product of both factors thus resembles the detection rate of a specific component. In the case of the corners, $\mathrm{o}_{\mathrm{C}}^{(C)}$ is consequently 1, and the amplitude a of the reference rate has been adjusted to our findings. The third factor $\mathrm{o}_{\mathrm{V}}^{(C)}$ is

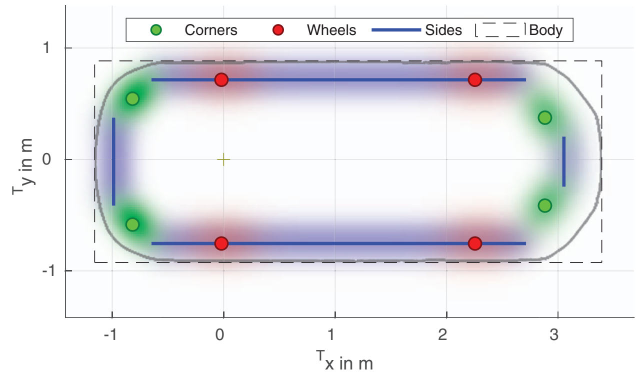


Fig. 3. The location of the components of the reflection model. The dots and lines denote the location of the components in the target frame; the gradients in the background the uncertainty of their location. The contour of a vehicle is overlayed for illustration purposes (gray).

a simple visibility constraint that checks if the corner is visible:

$$\mathrm{o}_{\mathrm{V}}^{(C)}(\mathbf{x}) = \begin{cases} 1 & \text{if adjacent vehicle sides are visible,} \\ 0 & \text{otherwise,} \end{cases} \tag{13}$$

which implies that a corner is considered visible if both adjacent sides of the vehicle are visible. The resulting detection rate $\mathrm{o}^{(C)}(\mathbf{x})$ of a corner is the product of all factors:

$$\mathrm{o}^{(C)}(\mathbf{x}) = \mathrm{o}_{\mathrm{R}}(r) \cdot \mathrm{o}_{\mathrm{C}}^{(C)}(\mathbf{x}) \cdot \mathrm{o}_{\mathrm{V}}^{(C)}(\mathbf{x}). \tag{14}$$

Table II
The Component Parametrization for Our Vehicle Model

Components	**Position** ${}^{\mathrm{T}}(x_C, y_C)^{(\cdot)}(\mathbf{x}_{\mathrm{ext}})$	**Position uncertainty** $\mathrm{Cov}\left({}^{\mathrm{T}}(x_C, y_C)^{(\cdot)}\right)$	**Detection rate** $\mathrm{o}^{(\cdot)}(\mathbf{x})$	**Doppler model**
Corner front {left, right}	$\begin{pmatrix} 0.65 \cdot l \\ \pm 0.25 \cdot w \end{pmatrix}$	$\mathrm{R}(\mp 45^\circ) \cdot \begin{pmatrix} (0.15\,\mathrm{m})^2 & (0\,\mathrm{m})^2 \\ (0\,\mathrm{m})^2 & (0.05\,\mathrm{m})^2 \end{pmatrix} \cdot \mathrm{R}(\mp 45^\circ)^\intercal$	Reference Rate × Base Rate (1) × Visibility Constraint	CTRV
Corner rear {left, right}	$\begin{pmatrix} -0.2 \cdot l \\ \pm 0.35 \cdot w \end{pmatrix}$	$\mathrm{R}(\pm 45^\circ) \cdot \begin{pmatrix} (0.15\,\mathrm{m})^2 & (0\,\mathrm{m})^2 \\ (0\,\mathrm{m})^2 & (0.05\,\mathrm{m})^2 \end{pmatrix} \cdot \mathrm{R}(\pm 45^\circ)^\intercal$	Reference Rate × Base Rate (0.66) × Visibility Constraint	none
Wheel front {left, right}	$\begin{pmatrix} 0.5 \cdot l \\ \pm 0.5 \cdot w \mp 0.15\,\mathrm{m} \end{pmatrix}$	$\begin{pmatrix} (0.2\,\mathrm{m})^2 & (0\,\mathrm{m})^2 \\ (0\,\mathrm{m})^2 & (0.1\,\mathrm{m})^2 \end{pmatrix}$		
Wheel rear {left, right}	$\begin{pmatrix} 0\,\mathrm{m} \\ \pm 0.5 \cdot w \mp 0.15\,\mathrm{m} \end{pmatrix}$	$\begin{pmatrix} (0.2\,\mathrm{m})^2 & (0\,\mathrm{m})^2 \\ (0\,\mathrm{m})^2 & (0.1\,\mathrm{m})^2 \end{pmatrix}$		
Side {left, right}	$\begin{pmatrix} -0.15 \cdot l \to 0.6 \cdot l \\ \pm 0.5 \cdot w \mp 0.15\,\mathrm{m} \end{pmatrix}$	$\begin{pmatrix} (0\,\mathrm{m})^2 & (0\,\mathrm{m})^2 \\ (0\,\mathrm{m})^2 & (0.05\,\mathrm{m})^2 \end{pmatrix}$	Reference Rate × Base Rate (0.29/1°) × Angular Width × Visibility Constraint × Scattering	CTRV
Side front	$\begin{pmatrix} 0.67 \cdot l \\ -0.125 \cdot w \to 0.125 \cdot w \end{pmatrix}$	$\begin{pmatrix} (0.05\,\mathrm{m})^2 & (0\,\mathrm{m})^2 \\ (0\,\mathrm{m})^2 & (0\,\mathrm{m})^2 \end{pmatrix}$		
Side rear	$\begin{pmatrix} -0.2 \cdot l \\ -0.15 \cdot w \to 0.15 \cdot w \end{pmatrix}$	$\begin{pmatrix} (0.05\,\mathrm{m})^2 & (0\,\mathrm{m})^2 \\ (0\,\mathrm{m})^2 & (0\,\mathrm{m})^2 \end{pmatrix}$		
Body	-	-	Reference Rate × Base Rate (0.11)	CTRV

The coordinates are given in the target frame T. The predicted state estimate $\mathbf{x}^\star$ provides the length l and the width w of the vehicle. $\mathrm{R}(\cdot)$ is the two-dimensional rotation matrix.

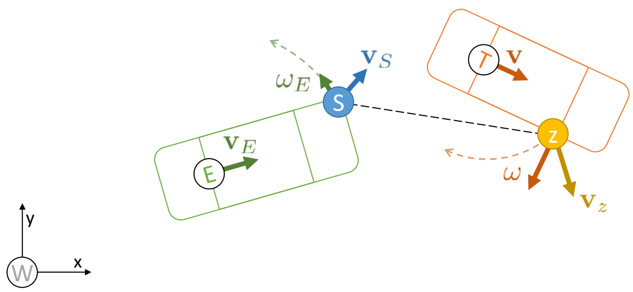


Fig. 4. The computation of the expected radial speed measurement. The velocity vector over ground of the radar sensor is the sum of the translational speed of the ego vehicle v_E and the rotary movement of the sensor caused by the yaw rate ω_E. The velocity vector of the requested point z on the target vehicle is similarly computed using v and ω. The radial speed measurement v_r of this point is the projection of the difference of both velocity vectors along the line of sight (dashed line).

The expected radial speed measurement is computed using the CTRV model [22]. Figure 4 accompanies the following calculation. Firstly, the velocity vector of the radar sensor over ground $\mathbf{v}_S$ is computed:

$$\mathbf{v}_S = \begin{pmatrix} \cos\varphi_E \cdot v_E - \omega_E \cdot (y_S - y_E) \\ \sin\varphi_E \cdot v_E + \omega_E \cdot (x_S - x_E) \end{pmatrix}. \tag{15}$$

This requires the longitudinal speed of the ego vehicle v_E, its heading in world coordinates φ_E, its yaw rate $\omega_E := \dot{\varphi}_E$, its pivot point (x_E, y_E), and the position of the sensor (x_S, y_S) in world coordinates. Secondly, the velocity vector $\mathbf{v}_z$ of the requested point $\mathbf{z}$, which lies on the target vehicle, is determined analogously:

$$\mathbf{v}_z(\mathbf{z}) = \begin{pmatrix} \cos\varphi \cdot v - \omega \cdot (y_\mathbf{z} - y) \\ \sin\varphi \cdot v + \omega \cdot (x_\mathbf{z} - x) \end{pmatrix}, \tag{16}$$

where the speed v, heading φ, yaw rate ω, and position of the target (x, y) are obtained from the target state estimate, while $(x_\mathbf{z}, y_\mathbf{z})$ are the world coordinates of the requested point. Thirdly, the orientation to the detection originating from the sensor Θ is determined:

$$\Theta(\mathbf{z}) = \operatorname{atan2}(x_\mathbf{z} - x_S, y_\mathbf{z} - y_S). \tag{17}$$

And fourthly, the difference of both velocity vectors is rotated to the radar frame:

$$v_r(\mathbf{z}) = \big(\cos\Theta(\mathbf{z}),\ \sin\Theta(\mathbf{z})\big) \cdot (\mathbf{v}_z(\mathbf{z}) - \mathbf{v}_S). \tag{18}$$

v_r returns the longitudinal velocity component or rather the radial speed. These equations are outlined in detail in [4, Section II.C].

B. Vehicle Wheels

The wheels of a vehicle are good reflectors, especially due to the metal rim and the suspension. We model the wheels as point targets, as their extents are also rather small. The rotating wheels cause radial speed measurements that do not match the body of the vehicle and cause the micro-Doppler effect. Thus, the radial speed measurement cannot be used for the kinematic state estimate of the vehicle. However, the radial speed measurement can be exploited to associate a nearby strong detection to a wheel: If the radial speed measurement mismatches the expected radial speed measurement of the body, then the wheel gains a high association probability. The component base rate $\mathrm{o}_\mathrm{C}^{(W)}$ is 0.66 according to our measurement analysis. The visibility constraint $\mathrm{o}_\mathrm{V}^{(W)}$, however, is more complex to model. For example, if the sensor is mounted at a typical low height, then the opposite wheels are in line of sight to it. Although the visible area of an opposite wheel can be computed, necessary parameters like the underbody height of the target are still unknown, and the estimation of it can be challenging. As a result, we reduce this problem to an empiric constant that corresponds to the average detection rate of opposite wheels according to our measurement analysis:

$$\mathrm{o}_\mathrm{V}^{(W)}(\mathbf{x}) = \begin{cases} 1 & \text{if the corresponding side is visible,} \\ 0.3 & \text{otherwise.} \end{cases} \tag{19}$$

The detection rate $\mathrm{o}^{(W)}$ is again the product of all factors:

$$\mathrm{o}^{(W)}(\mathbf{x}) = \mathrm{o}_\mathrm{R}(r) \cdot o_\mathrm{C}^{(W)}(\mathbf{x}) \cdot \mathrm{o}_\mathrm{V}^{(W)}(\mathbf{x}). \tag{20}$$

C. Vehicle Sides

The sides of a vehicle resemble the largest part of the shape of the vehicle. Therefore, they are a significant source of detections. Their extent no longer justifies a point target approximation. Especially, in static scenarios, where a specific part of a side has the highest reflectivity and causes nonuniformly distributed detections along the side, a bias occurs. It shifts the center, or mean, of the side toward that part. Instead, each point of the side has to be regarded as a potentially independent detection source. As a result, we consider each point of the line as a subcomponent of the side of the car, with each point having an independent measurement function to obtain the expected position and radial speed measurements. The detection rate $\mathrm{o}^{(S)}$ for the complete side consists of multiple factors. Figure 5 outlines the calculation of these factors. The first ones are again the reference rate $\mathrm{o}_\mathrm{R}(r)$ and the component base rate $\mathrm{o}_\mathrm{C}^{(S)}$. The latter must now be referenced to a certain angular width like 1° to consider the actual observed width. Multiplying this reference angular width with the actual observed angular width, ψ, then gives the final component base rate. The observed angular width, ψ, depends on the distance, the absolute length, and the orientation of the side. It is computed using the edge points of the side $\mathbf{A} = (x_A, y_A)$ and $\mathbf{B} = (x_B, y_B)$, which are defined in a counter-clockwise order around the center of the vehicle:

$$\psi = \big|\angle(\overrightarrow{\mathbf{SA}}, \overrightarrow{\mathbf{SB}})\big|. \tag{21}$$

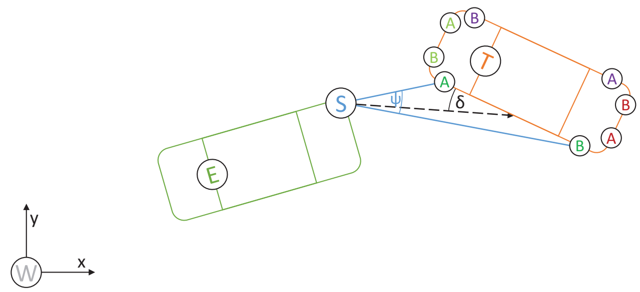


Fig. 5. The computation of the detection occurrence likelihood for the vehicle sides. The rear and right side of the target vehicle T are considered as visible because the sensor S is "right" of the respective vector $\overrightarrow{AB}$. The observed angular width ψ and the angle of incidence δ are drawn for the right side.

When measuring $o_C^{(S)}$, the vehicle side needs to precisely face the sensor. According to our evaluation, this factor is $0.29/1°$.

The next factor is the visibility constraint $q_v{}^{(S)}$, which checks if the side is oriented to the radar sensor. This is mathematically performed by computing the non-normalized signed distance $d_S^{(S)}$ of the radar sensor $\mathbf{S} = (x_S, y_S)$ toward the side:

$$d_S^{(S)} = (x_S - x_A) \cdot (y_B - y_A) - (y_S - y_A) \cdot (x_B - x_A)\,. \tag{22}$$

The boolean visibility constraint $o_V^{(S)}$ is then given by a sign check:

$$o_V^{(S)}(\mathbf{x}) = \begin{cases} 1 & \text{if } d_S^{(S)} > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{23}$$

The last factor represents the *scattering effect*. The reflections of the radar waves are scattered depending on the angle of incidence δ:

$$\delta = \operatorname{atan2}(y_B - y_A, x_B - x_A) - \angle(\overrightarrow{\mathbf{SM}}, \overrightarrow{\mathbf{AB}}), \tag{24}$$

where the point $\mathbf{M}$ is the midpoint of the side and used as approximative reference:

$$\mathbf{M} = 1/2(\overrightarrow{\mathbf{A}} + \overrightarrow{\mathbf{B}}). \tag{25}$$

This approximation is required as each point of the line has a different angle of incidence. It is fully sufficient for vehicles that are not in the immediate vicinity. Later, in the state update, $\overrightarrow{\mathbf{SM}}$ can also be replaced by the actual orientation of the radar toward the detection.

The steeper the angle between the vehicle side and the radar, the more signal power is scattered and the less signal power is reflected back to the sensor. This damping factor $o_S^{(S)}$ is provided by a model proposed in [15] and [3]:

$$o_S^{(S)}(\mathbf{x}) = \sin(\delta)^2. \tag{26}$$

The detection rate finally results in

$$o^{(S)}(\mathbf{x}) = o_R(r) \cdot o_C^{(S)}(\mathbf{x}) \cdot \frac{\psi}{1°} \cdot o_V^{(S)}(\mathbf{x}) \cdot o_S^{(S)}(\mathbf{x}). \tag{27}$$

D. Vehicle Body

Finally, a portion of detections is caused by arbitrary, model-specific parts of the vehicle body. It is not possible to perform a position update of the estimate as the origin of the detection is unknown. However, a kinematic update is conceivable for the current position of the detection. We derived the component base rate $o^{(B)}$ of the vehicle body from the measurements (i.e., 0.11):

$$o^{(B)}(\mathbf{x}) = o_R(r) \cdot o_C^{(B)}(\mathbf{x}). \tag{28}$$

At this point, all relevant components of a vehicle have been abstracted to a set of generic functions.

V. EXPECTED SPATIAL DETECTION LIKELIHOOD

This section depicts the computation of the expected spatial detection likelihood based on the generic component descriptions. The expected spatial detection likelihood serves as a measurement function for an arbitrary object state $\mathbf{x}$ and indicates the expected number of detections for any point in the measurement space $\mathbf{z}$. Thereby, it also takes into account the expected state uncertainty $\mathbf{P}$. The spatial detection likelihood can also be interpreted as the detection rate or frequency for a given point, or as a probabilistic detection density. The spatial sum of the detection likelihood corresponds to the expected number of detections the complete object presumably invokes.

The following equations are given for a single time step. Hence, the corresponding indices are omitted for the sake of simplicity. As a prerequisite, the algorithm requires the component locations $(x_C, y_C)^{(\cdot)}$ to be transformed from target coordinates T to world coordinates W. The mean transformation is given by

$$^{\mathrm{W}}\begin{bmatrix} x_C \\ y_C \end{bmatrix}^{(\cdot)} = \begin{bmatrix} x^\star \\ y^\star \end{bmatrix} + \mathrm{R}(\varphi^\star) \cdot {}^{\mathrm{T}}\begin{bmatrix} x_C \\ y_C \end{bmatrix}^{(\cdot)}, \tag{29}$$

and its uncertainty transformation by

$$\mathrm{Cov}\left({}^{\mathrm{W}}(x_C, y_C)^{(\cdot)}\right) = \mathrm{R}(\varphi^\star) \cdot \mathrm{Cov}\left({}^{\mathrm{T}}(x_C, y_C)^{(\cdot)}\right) \cdot \mathrm{R}(\varphi^\star)^\intercal. \tag{30}$$

In the following, we perform all computations in world coordinates and omit the coordinate frame index W and time indices to simplify the formal representation.

The spatial detection likelihood indicates the chance of obtaining a detection for any desired point in the measurement space. The likelihood is computed for each component and depends on the object state, the position of the component, its extent model, and also on the uncertainties of both the state and the measurement.

A. Vehicle Corners

The expected measurement vector of a corner consists of a position and radial speed measurement:

$$\mathbf{y}^{\star(C)} = \begin{bmatrix} x_C^{(C)}(\mathbf{x}^\star) \\ y_C^{(C)}(\mathbf{x}^\star) \\ v_r\left((x_C, y_C)^{(C)}(\mathbf{x}^\star)\right) \end{bmatrix}. \quad (31)$$

The measurement matrix $\mathbf{C}^{(C)}$ for a corner can be determined using linearization:

$$\mathbf{C}^{(C)} = \left.\frac{\partial \begin{bmatrix} x_C^{(C)}(\mathbf{x}) \\ y_C^{(C)}(\mathbf{x}) \\ v_r\left((x_C, y_C)^{(C)}(\mathbf{x})\right) \end{bmatrix}}{\partial \mathbf{x}}\right|_{\mathbf{x}^\star}. \quad (32)$$

Subsequently, the innovation covariance matrix $\mathbf{S}^{(C)}$ yields

$$\mathbf{S}^{(C)} = \mathbf{C}^{(C)}\mathbf{P}^\star\mathbf{C}^{(C)\mathsf{T}} + \begin{bmatrix} \mathrm{Cov}\left((x_C, y_C)^{(C)}\right) + \mathbf{R}_{\mathbf{xy}} & 0 \\ 0 & \sigma_{\dot{r}}^2 \end{bmatrix}, \quad (33)$$

which treats the location uncertainty $\mathrm{Cov}((x_C, y_C)^{(C)})$ of the corner as an additive measurement uncertainty. At this point, the spatial detection likelihood caused by a corner $\gamma^{(C)}(\mathbf{z})$ can be computed using the Gaussian distribution $\mathcal{N}(\cdot)$:

$$\begin{aligned}\gamma^{(C)}(\mathbf{z}) &= \mathrm{o}^{(C)}(\mathbf{x}^\star) \cdot \mathcal{N}\left(\mathrm{x}=\mathbf{z}, \mu=\mathbf{y}^{\star(C)}, \sigma^2=\mathbf{S}^{(C)}\right) \\ &= \frac{\mathrm{o}^{(C)}(\mathbf{x}^\star)}{\sqrt{(2\pi)^3 \det\left(\mathbf{S}^{(C)}\right)}} \cdot \\ &\quad \exp\left(-\frac{1}{2}\left(\mathbf{z}-\mathbf{y}^{\star(C)}\right)^{\mathsf{T}} \mathbf{S}^{(C)^{-1}} \left(\mathbf{z}-\mathbf{y}^{\star(C)}\right)\right). \end{aligned} \quad (34)$$

B. Vehicle Wheels

The spatial detection likelihood of the wheels is similar to the corners, but the micro-Doppler effect prevents the usage of the radial speed measurement. The expected measurement $\mathbf{y}^{\star(W)}$ is given by

$$\mathbf{y}^{\star(W)} = \begin{bmatrix} x_C^{(W)}(\mathbf{x}^\star) \\ y_C^{(W)}(\mathbf{x}^\star) \end{bmatrix}, \quad (35)$$

and the measurement matrix $\mathbf{C}^{(W)}$ is given by

$$\mathbf{C}^{(W)} = \left.\frac{\partial \begin{bmatrix} x_C^{(W)}(\mathbf{x}) \\ y_C^{(W)}(\mathbf{x}) \end{bmatrix}}{\partial \mathbf{x}}\right|_{\mathbf{x}^\star}. \quad (36)$$

The innovation covariance matrix $\mathbf{S}^{(W)}$ is

$$\mathbf{S}^{(W)} = \mathbf{C}^{(W)}\mathbf{P}^\star\mathbf{C}^{(W)\mathsf{T}} + \mathrm{Cov}\left((x_C, y_C)^{(W)}\right) + \mathbf{R}_{\mathbf{xy}}. \quad (37)$$

Therefore, the spatial detection likelihood can be described by

$$\gamma^{(W)}(\mathbf{z}) = \mathrm{o}^{(W)}(\mathbf{x}^\star) \cdot \mathcal{N}\left(\mathrm{x}=\mathbf{z}, \mu=\mathbf{y}^{\star(W)}, \sigma^2=\mathbf{S}^{(W)}\right). \quad (38)$$

C. Vehicle Sides

As discussed, the length of the vehicle sides demands a more sophisticated handling than the approximation as a point target. Instead, we model a vehicle side as a line. Each point of the line can be the possible source of a detection. As a result, the uncertainties of the state and the measurement reveal a subordinated, continuous association ambiguity for a given detection, as there is a span of possible point sources for a given detection. We aim for a continuous approach [5, Section II.B] to solve the association ambiguity: splitting the line into segments would result in more runtime efforts and only attenuate the bias effect that is evoked by discretized sampling points. To begin, we consider a point $\mathbf{s}^{(S)}(u) \in S$, $u \in [0; 1]$. Its parametrization can be formally represented as

$$\mathbf{s}^{(S)}(u) = \begin{bmatrix} x_A(\mathbf{x}^\star) + u \cdot (x_B(\mathbf{x}^\star) - x_A(\mathbf{x}^\star)) \\ y_A(\mathbf{x}^\star) + u \cdot (y_B(\mathbf{x}^\star) - y_A(\mathbf{x}^\star)) \end{bmatrix}, \quad (39)$$

by utilizing both end points of the side $(x_A(\mathbf{x}^\star), y_A(\mathbf{x}^\star))$ and $(x_B(\mathbf{x}^\star), y_B(\mathbf{x}^\star))$ in world coordinates W. This allows for the denotation of the expected measurement vector as

$$\mathbf{y}^{\star(s)}(u) = \begin{bmatrix} \mathbf{s}^{(S)}(u) \\ v_r\left(\mathbf{s}^{(S)}(u)\right) \end{bmatrix}, \quad (40)$$

the measurement matrix as

$$\mathbf{C}^{(s)}(u) = \left.\frac{\partial \begin{bmatrix} \mathbf{s}^{(S)}(u) \\ v_r(\mathbf{s}^{(S)}(u)) \end{bmatrix}}{\partial \mathbf{x}}\right|_{\mathbf{x}^\star}, \quad (41)$$

the innovation covariance matrix as

$$\begin{aligned}\mathbf{S}^{(s)}(u) &= \mathbf{C}^{(s)}(u)\mathbf{P}^\star\mathbf{C}^{(s)\mathsf{T}}(u) \\ &\quad + \begin{bmatrix} \mathrm{Cov}\left((x_C, y_C)^{(S)}\right) + \mathbf{R}_{\mathbf{xy}} & 0 \\ 0 & \sigma_{\dot{r}} \end{bmatrix}, \end{aligned} \quad (42)$$

and finally, the spatial detection likelihood as

$$\gamma^{(s)}(u, \mathbf{z}) = \mathrm{o}^{(S)}(\mathbf{x}^\star) \cdot \mathcal{N}\left(\mathrm{x}=\mathbf{z}, \mu=\mathbf{y}^{\star(s)}(u), \sigma^2=\mathbf{S}^{(s)}\right). \quad (43)$$

The spatial detection likelihood invoked by the complete vehicle side can be computed by summing up the detection likelihoods of all the points:

$$\gamma^{(S)}(\mathbf{z}) = \int_0^1 \gamma^{(s)}(u, \mathbf{z}) \, \mathrm{d}u. \quad (44)$$

This integral is known as *stick model* in the literature. Some simplifications of the stick model and the following integrals yield short closed functions. As the computation of the expected radial speed measurement is rather complex, we approximate it by a linear function. This approximation is only used for the association steps and uses both end points as sampling points. The measurement matrix $\mathbf{C}^{(s)}(u)$ is different for all points, mainly because of the altering impact of the uncertainty of the yaw of the state estimate. Due to its symmetric characteristic, we consider a static innovation covariance matrix, which is either sampled for the center of the line or for the nearest point of the line from the detection.

D. Vehicle Body

The last component class represents arbitrary detections on the body of the vehicle. As the actual source is unknown, a position update is not feasible. However, the expected radial speed measurement can be computed for any point $\mathbf{z}$. This allows for purely kinematic association hypotheses and state updates. The latter is viable by filtering the radial speed measurement at the position of the measured detection. The measurement vector for the body component consequently consists only of the radial speed measurement $\dot{r}$:

$$\mathbf{y}^{(B)} = \dot{r}, \tag{45}$$

while the corresponding expected measurement is given by

$$\mathbf{y}^{\star(B)} = v_r(\mathbf{z}). \tag{46}$$

By considering the partial derivative, the body measurement matrix $\mathbf{C}^{(B)}$ yields

$$\mathbf{C}^{(B)} = \left.\frac{\partial v_r(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}^\star}, \tag{47}$$

and the innovation covariance matrix $\mathbf{S}^{(B)}$ yields

$$\mathbf{S}^{(B)} = \mathbf{C}^{(B)}\mathbf{P}^\star\mathbf{C}^{(B)\intercal} + \sigma_{\dot{r}}^2. \tag{48}$$

With these equations, we can compute the detection likelihood $\gamma^{(B)}$:

$$\gamma^{(B)}(\mathbf{z}) = \mathrm{o}^{(B)}(\mathbf{x}^\star)\cdot\mathcal{N}\left(\mathrm{x}{=}\mathbf{z}, \mu{=}\mathbf{y}^{\star(B)}, \sigma^2{=}\mathbf{S}^{(B)}\right), \tag{49}$$

which indicates the likelihood that the body component is the source of the detection.

E. Clutter (Optional)

Similar to probabilistic data association filter (PDAF) applications, a clutter hypothesis can be added to the association problem. Clutter is caused by false positive detection measurements and part of the sensor model (Section III). The clutter likelihood $\gamma^{(0)}$ can be modeled with a Poisson distribution. Its parameters usually depend on the distance and the environmental complexity and are firmware-specific. They can be determined with an appropriate measurement analysis. Clutter is not part of the object model itself. However, for single-object tracking applications, it can be interpreted as an additional virtual component that does not invoke a state innovation.

Figure 6 shows the spatial detection likelihoods for different observation angles. Besides, the superposition of all expected spatial detection likelihoods $\gamma^{(J)}(\mathbf{z}) = \sum_{j\in J}\gamma^{(j)}(\mathbf{z})$ is illustrated and compared with the measurement data. As the measurement data can only be visualized for a span of observation angles, the superposed spatial likelihoods are sampled and averaged over this span to match the data visualization. Note that Fig. 6 only shows the position components (x, y) of the measurement space vector $\mathbf{z} = (x, y, \dot{r} = 0)$. A dynamic scenario is shown in Fig. 1 and rendered in the full measurement space.

At this point, our proposed (spatial) measurement function for an extended object is available. It can be used in a Bayes filter like a particle filter to judge state hypotheses and thereby to estimate the state of an object.

VI. DETECTION-TO-COMPONENT ASSOCIATION PROBABILITY

This section outlines the computation of the association probability $\beta^{(j)}$ of a single detection measurement $\mathbf{y}$ toward any component j. This is done by normalizing their detection likelihoods $\gamma(\cdot)$ for the given detection $\mathbf{y}$:

$$\beta^{(j)} = \frac{\gamma^{(j)}(\mathbf{y})}{\sum_{k\in J}\gamma^{(k)}(\mathbf{y})}. \tag{50}$$

Figure 7 illustrates the prior detection likelihoods and the association hypotheses for an example target state estimate with a realistic state and measurement uncertainty. The target is positioned at ${}^{\mathrm{W}}(0\,\mathrm{m}, 0\,\mathrm{m})$ and parked at a heading of 30°. The radar sensor is positioned at ${}^{\mathrm{W}}(0\,\mathrm{m}, -10\,\mathrm{m})$. At this distance, the (Euclidean) lateral measurement noise of the radar sensor is significantly higher than the longitudinal one. First, the priors are computed. The visibility constraints predict visibility for the rear and right vehicle sides, the rear right corner, and all wheels. The a priori detection likelihood of the right vehicle side is higher than the detection likelihood of the rear side because the angle of incidence causes a significantly higher scattering effect at the rear side. The sum of all a priori detection likelihoods is approximately two, i.e., the measurement model expects two detections to be obtained in this scenario. The prior spatial detection likelihood takes all uncertainties into account and predicts the occurrence of detections in the measurement space. It is visualized in the background (gradients). As it is not possible to print the three-dimensional detection likelihood, the gradients are rendered for the sectional plane given by $\mathbf{z} = (x^\star, y^\star, \dot{r}^\star = \dot{r})$, i.e., the plane in the z-axis of Fig. 1 that corresponds to the actually measured radial speed. This foreknowledge about the detection measurement at this point is limited to this illustrative purpose. According to the illustrated spatial detection likelihood, there is a high probability that these are located on the

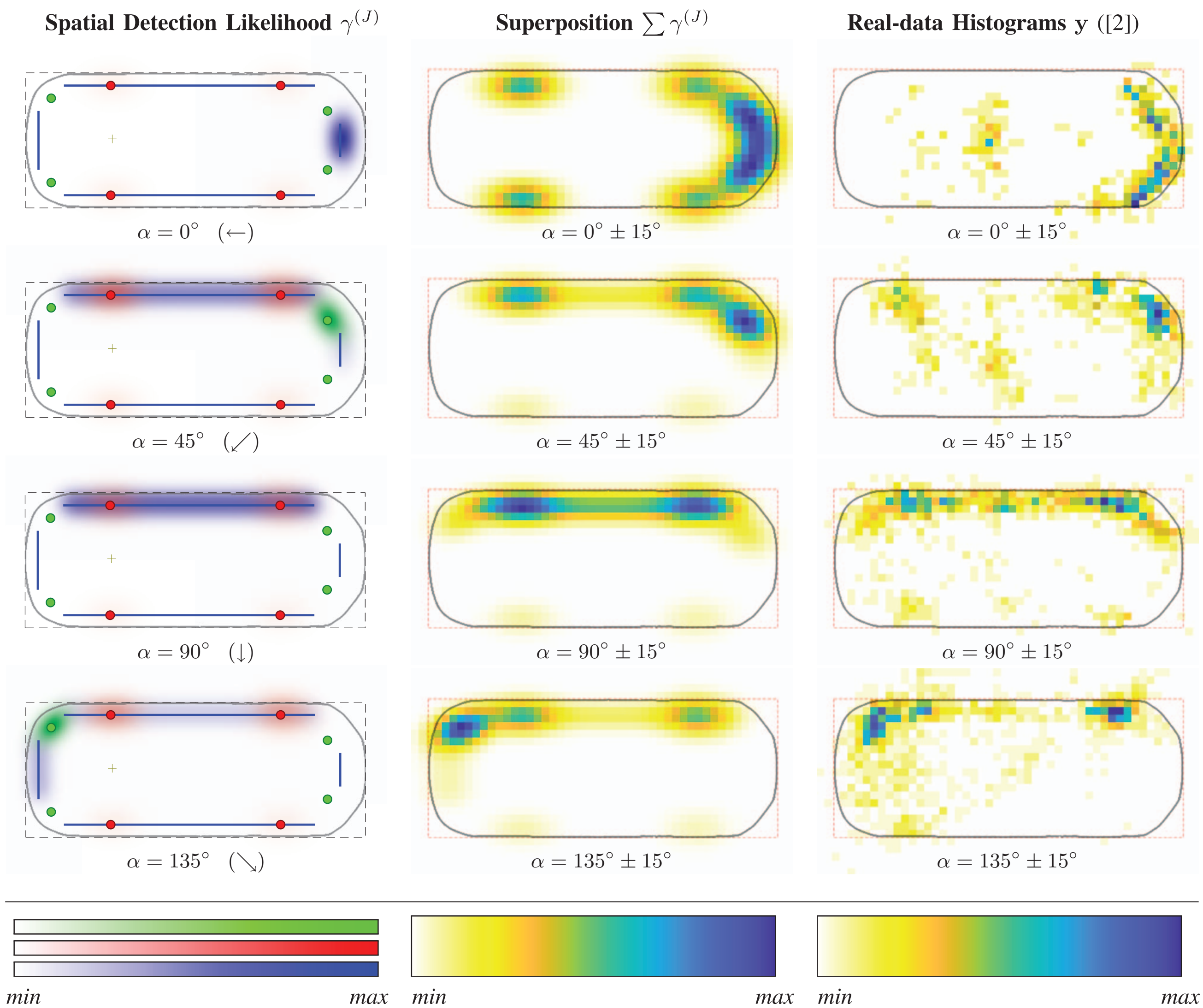


Fig. 6. Comparison of the approximated model and the measurement data: The left column shows the spatial prior detection likelihood for various observation angles α. The middle column shows their superposition for a certain range of observation angles, for which the measurement data on the right column have been respectively recorded. The resolutions are adjusted. The colors of the spatial detection likelihood plots correspond to Fig. 3. The data histograms are collected over several minutes.

rear right corner or on the right vehicle side. The likelihood that a detection is measured at the back side or the rear right wheel is lower.

As the illustrated detection measurement (blue point) is received, the association probabilities can be computed. The detection results in a high association probability of the right side, a moderate probability of the back side, but a low probability of the rear right corner (primarily due to the lower longitudinal measurement uncertainty). The shown orange arrows represent the association hypotheses. In case of the sides, they point to their mean origin point $\bar{\mathbf{s}}^{(S)}$. This point is the average of all points on the side, but weighted by their individual association likelihood:

$$\bar{u} = \left(\int_0^1 u \cdot \gamma^{(s)}(u, \mathbf{y}) \, \mathrm{d}u \right) \Bigg/ \left(\int_0^1 \gamma^{(s)}(u, \mathbf{y}) \, \mathrm{d}u \right), \quad (51)$$

$$\bar{\mathbf{s}}^{(S)} = \begin{bmatrix} x_A(\mathbf{x}^\star) + \bar{u} \cdot (x_B(\mathbf{x}^\star) - x_A(\mathbf{x}^\star)) \\ y_A(\mathbf{x}^\star) + \bar{u} \cdot (y_B(\mathbf{x}^\star) - y_A(\mathbf{x}^\star)) \end{bmatrix}. \quad (52)$$

The obtained mean origin point $\bar{\mathbf{s}}^{(S)}$ can be used to re-compute the measurement matrix $\mathbf{C}^{(s)}(u)$ in a recursive approach. As the lateral innovation uncertainty is higher than the longitudinal uncertainty, the mean origin points of the sides are mainly laterally shifted from the detection measurement.

Figure 8 introduces dynamics in the scenario: The target drives in a curve to the left. The radial speed measurement is set in a manner that it matches the expected radial speed measurement of the rear side. As the rear slightly moves toward the sensor in a left curve, the radial speed measurement is negative. Given the low uncertainty of the radial speed measurement, the detection is now associated with the rear side with high significance. Additionally, the hypothesis that the detection originates from the wheel emerges as its

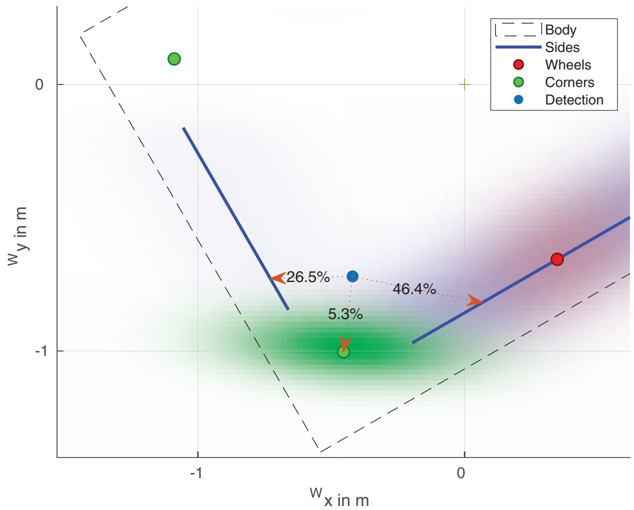


Fig. 7. Association probabilities of a static scenario. The orange arrows denote the association probabilities. The probabilities for the body (18.0%) and clutter (3.6%) are not shown.

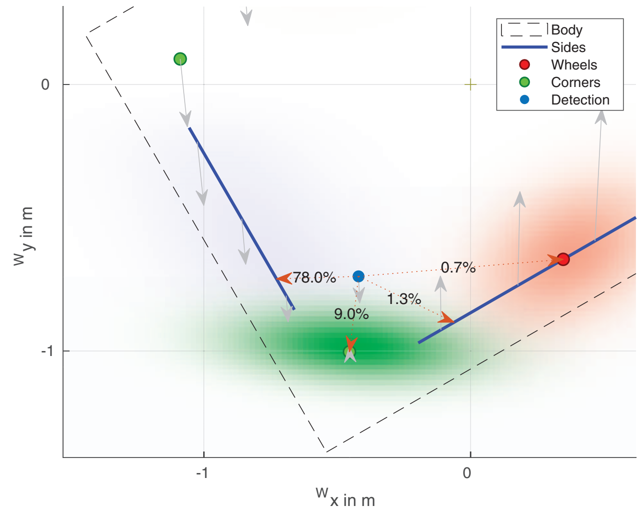


Fig. 8. Association probabilities of a dynamic scenario. The probability for the body is ~0% and for the clutter is 10.9%. The gray arrows denote the radial speed measurement of the detection and the expected radial speed measurements of all components ($1\,\mathrm{m} \widehat{=} 1\,\mathrm{ms}^{-1}$).

association ignores the mismatching radial speed differences due to the micro-Doppler effect. The estimated mean origin of the right side is shifted to the left to better match the radial speed measurement, but it still loses any significant association likelihood.

The association probabilities $\beta^{(j)}$ can now be used to obtain a probabilistic indication about the origin of a detection.

VII. STATE UPDATE

This section describes the filter principle and its actual implementation to update the state and uncertainty of a single-object state hypothesis $\mathbf{x}$.

A. Principle

Each component is a possible source of a detection. Especially when considering the uncertainty of both measurement and state, a given detection could originate from multiple components. A particle filter that matches the complete spatial detection frequency or a multiple hypothesis tracking (MHT)-adapted approach that tracks the associations of the detections to the components over time is not feasible in an application where a multi-hypotheses tracking is run upstream. A simple maximum a posteriori estimate, or hard association, though, does not establish a robust tracking due to the high ambiguity of the association problem. We aim for a soft association approach, which represents a suitable compromise according to our findings. The association ambiguities are resolved probabilistically and are still encased in a Gaussian state formulation. Association uncertainties are thereby incorporated in the state uncertainty. The utilized association and tracking algorithm has been developed previously [5] as preparation for this work. It shares its basic principles with the PDAF [1] and is briefly stated in the following.

Every component j has a state-dependent association likelihood $\gamma^{(j)}$. It denotes the presumption that component j has caused a given detection measurement. The absolute association probability, $\beta^{(j)}$, is determined by computing the association likelihoods for all components J and by normalizing them, as done in equation (50). At this point, clutter measurements are not yet considered. As each component description correlates the object state with the component, it can also provide a state update $\hat{\mathbf{x}}^{(j)}$ of the predicted target state $\mathbf{x}^\star$. This update is conditioned on the assumption that the detection is actually caused by the component j:

$$\hat{\mathbf{x}}^{(j)} = \mathbf{x}^\star + \mathbf{K}^{(j)} \cdot \left(\mathbf{y} - \mathbf{y}^{\star(j)}\right), \tag{53}$$

where $\mathbf{K}^{(j)}$ is the Kalman gain of component j, $\mathbf{y}$ is the detection measurement, and $\mathbf{y}^{\star(j)}$ is the expected measurement if component j is assumed to be the origin of the detection. In a last step, the conditional state updates $\hat{\mathbf{x}}^{(j)}$ are fused according to their association probabilities $\beta^{(j)}$ to obtain the updated state $\hat{\mathbf{x}}$:

$$\hat{\mathbf{x}} = \sum_{j\in J} \beta^{(j)} \cdot \hat{\mathbf{x}}^{(j)}. \tag{54}$$

The updated state uncertainty $\hat{\mathbf{P}}$ is calculated similarly by

$$\hat{\mathbf{P}} = \sum_{j\in J} \beta^{(j)} \left((\mathbf{1} - \mathbf{K}^{(j)}\mathbf{C}^{(j)})\mathbf{P}^\star + \underline{(\hat{\mathbf{x}}^{(j)} - \hat{\mathbf{x}})(\hat{\mathbf{x}}^{(j)} - \hat{\mathbf{x}})^\intercal} \right) \tag{55}$$

and depends on the predicted state uncertainty $\mathbf{P}^\star$ and measurement matrices $\mathbf{C}^{(j)}$ for all components $j \in J$. The term marked with a dashed underline represents the uncertainty of the association, the so-called *spread* *of means*. This term carries the information how certain the association search is. It increases according to the cardinality, the likelihood, and the impact of alternative association hypotheses.

This probabilistic association requires a far lower computing expense than multi-hypotheses trackers, which resolve the combinatorial association problem over time. The downside of probabilistic associations is that even wrong associations are filtered in with a certain weight. In this application, significant association probabilities are only invoked by components that are of comparable likelihood to have caused the detection. Such components are mostly close together, and their state updates are similar as they are part of a rigid body. This is a major difference to PDAF applications where multiple, independent object tracks are updated with a single measurement.

B. Implementation

In the following, a Kalman filter update of the target state, considering a single-object hypothesis, is performed.

The update of the target state is composed of update steps for each single component. According to the principle of the probabilistic origin association, each individual component update step is performed in the assumption that the respective component is the origin of the given detection, regardless of its actual association probability. For every component $j \in J$, the Kalman gain $\mathbf{K}^{(j)}$ is computed as

$$\mathbf{K}^{(j)} = \mathbf{P}^{\star}\mathbf{C}^{(j)\mathsf{T}}\mathbf{S}^{(j)-1} \tag{56}$$

and utilized for the component-wise state updates $\hat{\mathbf{x}}^{(j)}$ according to equation (53), and subsequently for the fused posterior state estimate $\hat{\mathbf{x}}$ according to equation (54) and the posterior state uncertainty $\hat{\mathbf{P}}$ according to equation (55). While the state updates $\hat{\mathbf{x}}^{(j)}$ can be directly obtained for the other components, the vehicle sides require a more elaborate treatment. Their native posterior state estimates $\hat{\mathbf{x}}^{(S)}$ yield

$$\begin{aligned}\hat{\mathbf{x}}^{(S)} &= \int_0^1 \beta^{(s)}(u) \cdot \hat{\mathbf{x}}^{(s)}(u)\ \mathrm{d}u \\ &= \int_0^1 \beta^{(s)}(u) \cdot \left(\mathbf{x}^{\star} + \mathbf{K}^{(s)}(u) \cdot \left(\mathbf{y} - \mathbf{y}^{\star(s)}(u)\right)\right)\ \mathrm{d}u.\end{aligned} \tag{57}$$

The recursive approximation of the measurement matrix $\mathbf{C}^{(s)}(u)$ for a static replacement $\mathbf{C}^{(s)}(\bar{\mathbf{s}}^{(S)})$, previously discussed in Section VI, simplifies the posterior state estimate to

$$\hat{\mathbf{x}}^{(S)} = \mathbf{x}^{\star} + \mathbf{K}^{(s)}(\bar{u}) \cdot \left(\mathbf{y} - \underline{\int_0^1 \beta^{(s)}(u) \cdot \mathbf{y}^{\star(s)}(u)\ \mathrm{d}u}\right), \tag{58}$$

where the dashed underlined term represents the mean of the origins. This mean has already been computed in equation (51) and yields

$$\hat{\mathbf{x}}^{(S)} = \mathbf{x}^{\star} + \mathbf{K}^{(s)}(\bar{u}) \cdot \left(\mathbf{y} - \mathbf{y}^{\star(s)}(\bar{u})\right). \tag{59}$$

This approximation implies that the Kalman gain of the expected mean origin is applied to the nearby, less likely origins in a symmetric manner. Therefore, the native posterior state uncertainty of a vehicle side $\hat{\mathbf{P}}^{(S)}$ is given by

$$\begin{aligned}\hat{\mathbf{P}}^{(S)} = \int_0^1 \beta(u)\Big(&\left(\mathbf{1} - \mathbf{K}^{(s)}(u)\mathbf{C}^{(s)}(u)\right)\mathbf{P}^{\star} \\ &+ \left(\hat{\mathbf{x}}^{(s)}(u) - \hat{\mathbf{x}}^{(S)}\right)\left(\hat{\mathbf{x}}^{(s)}(u) - \hat{\mathbf{x}}^{(S)}\right)^{\mathsf{T}}\Big)\ \mathrm{d}u\end{aligned} \tag{60}$$

and can be simplified with the same approximation $\mathbf{C}^{(s)}(u) \approx \mathbf{C}^{(s)}(\bar{u})$ to obtain a closed equation, albeit too long to be printed here. Details on the analytic solution can be found in the supplied MATLAB code. Similar to the discrete association problem, this term incorporates the uncertainty of the association search into the resulting innovation uncertainty. In this way, high state or measurement uncertainties increase the possible association range and are—in contrast to pure greedy decision approaches—probabilistically resolved.

Another implementation issue concerns the computation of the expected radial speed measurement. In the association step, it is computed for the mean position of the respective component (x_C, y_C). Its advantage is the improved search for the origin of a detection by comparing it with the precise radial speed measurement. However, the actual origin of the detection can be located anywhere on the extent of the component; it is spatially distributed according to its position mean and uncertainty parametrization in Table II. Depending on the size of the respective extent, this discrepancy might cause a pseudo-systematic bias in the state update, especially when perceiving the component repeatedly from a similar angle. An alternative is the usage of the measured position of the detection (x_D, y_D): It is instead subject to (zero-mean) measurement noise. The choice depends on the magnitude of the estimated uncertainties, the measurement noise and the extent of the components. We have performed an ablation study and gained the result that both variants perform almost identical, primarily because the Gaussian extents are minor relative to the Doppler gradient [4]. We utilize the second option for (and only for) the state update to achieve higher generality with respect to extent sizes. This requires the reprocessing of equations (31) and (40) to use $v_r\big((x_D, y_D)\big)$, and subsequently the reprocessing of their respective measurement and innovation covariance matrices described by equations (32), (33), (41), and (42).

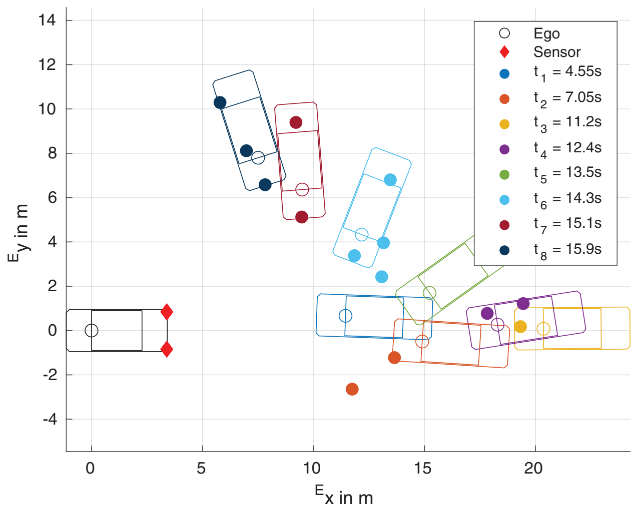


Fig. 9. Country road trailing: Exemplary measurement time steps and their respective true target positions *in the ego frame*. The blank circles denote the reference points of the ego vehicle (black) and the target vehicle (colored according to the timestamp). The filled circles denote the respective detection measurements. No detections were obtained in t_1 and t_5.

VIII. SINGLE-OBJECT TRACKING PERFORMANCE

This section focuses on single-object tracking and illustrates three different tracking scenarios. The measurement data has been recorded with low-resolution radar sensors, which are mounted on the corners of the ego vehicle. We utilize a RTK-GNSS/IMU-based ground truth with centimeter-level accuracy for both the gating of the radar detections (within a radius of 4 m to the center of the target) and the evaluation of the tracking performance. Figure 9 shows the measured detections and the ground-truth-provided true object state of some representative measurement frames of the first curve of the first scenario. This illustration reveals the challenge that the tracking algorithm has to tackle. On the one hand, the number of detections is low, the detections are generated at unknown positions and are subject to significant measurement noise. On the other hand, the dynamic variables of the object state can change abruptly. Wrong associations would directly impair the tracking robustness. The association algorithm primarily exploits the statistical detection characteristic, provided by the object model, and the radial speed measurements to solve the association problem. These difficulties should be considered when assessing the resulting tracking performance.

All scenarios use the same parametrization. The process noise of the CTRV model has been obtained by the inspection of a larger dataset and regards slight model inconsistencies concerning unpaved roads, slopes, and varying driving styles. We parameterize it by

$$\sigma^2_{\text{kin}} = \text{diag}\big([(4.5\,\text{cm})^2, (4.5\,\text{cm})^2, \tag{61}$$

$$(1.1^\circ)^2, (0.67\,\text{ms}^{-1})^2, (6.3^\circ\text{s}^{-1})^2]\big). \tag{62}$$

There is no process noise modeled for the extent state model. The initial CTRV position is roughly set to the first encountered measurement. All kinematic means are zero. All CTRV state parameters are initialized as extremely uncertain. The initialization of the extent state, though, depends on the application. In general, the low number of detections obtained in the usual observation time, as in urban scenarios, does not permit a very precise extent estimation. In such applications, the extent state should be initialized with an average extent state (like $\mathbf{x}^{\mathsf{T}}_{\text{ext},0} = [4.85\,\text{m}, 1.85\,\text{m}]$) and with a low uncertainty. Longer observation times, as given in the presented scenarios, render extent estimation feasible. The initial extent is set to $\mathbf{x}^{\mathsf{T}}_{\text{ext},0} = [4.7\,\text{m}, 1.75\,\text{m}]$. The length of the utilized target vehicles exceeds this by up to $\sim$0.6 m: This initial mismatch additionally challenges the component association search. The initial extent uncertainty is set to $\sigma^2_{\text{l},0} = 0.1\ \text{m}^2$ and $\sigma^2_{\text{w},0} = 0.015\ \text{m}^2$. Especially the variety of the width among typical vehicles is obviously bounded by regulations [21].

To deal with clutter in the single-object tracking (without a track management that handles clutter itself), we utilize the clutter hypothesis from Section V-E. We set $\gamma^{(0)} = 0.01$. This implies that roughly 1% of all detection measurements in immediate proximity of the target are clutter. This value is conservatively modeled without dependency to distance or signal strength, as such factors are already regarded in the clutter suppression of the sensor firmware. The clutter hypothesis has a certain "association" probability in equation (50) depending on the matching of a detection measurement to the real components. The clutter is then ignored in the subsequent state updates [equations (54) and (55)]. Consequently, the clutter detection is resolved probabilistically. A detection measurement that does not match any component at all (after consideration of all uncertainties), will not invoke a state update.

A. Country Road Trailing

In this scenario, the ego vehicle follows the target (a mid-class sedan) in a winding round trip. Clutter measurements are obtained from vegetation on the road side. The short-range mass-market, 77 GHz radar sensors utilized in this scenario have a substantial lateral measurement noise. A sensor is mounted at each corner of the vehicle, although only the front two sensors perceive the target. Each sensor provides measurements at a rate of roughly 20 Hz.

Figure 10 outlines the path of the ego vehicle (both by estimate and ground truth) in world coordinates, while Fig. 11 illustrates the accumulation of detection measurements and position estimates over time in the target frame (based on the ground truth). The lateral measurement noise also manifests itself in a lateral position estimate error. Moreover, Fig. 12 provides the yaw estimate over time, while Figs. 13 and 14 outline the estimation errors of the dynamic states. Their estimation

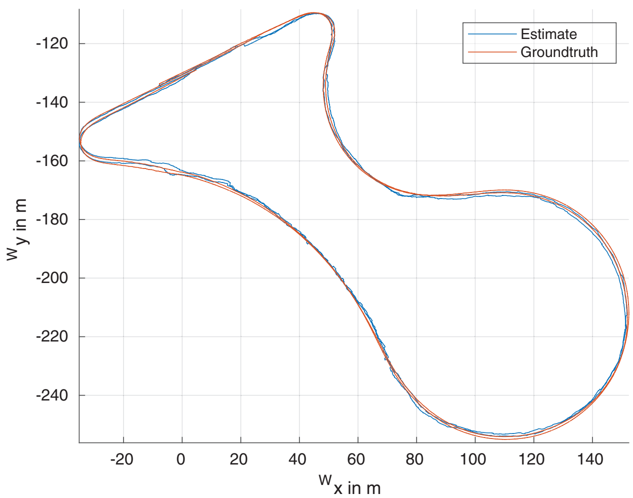


Fig. 10. Country road trailing: Position estimate in world coordinates.

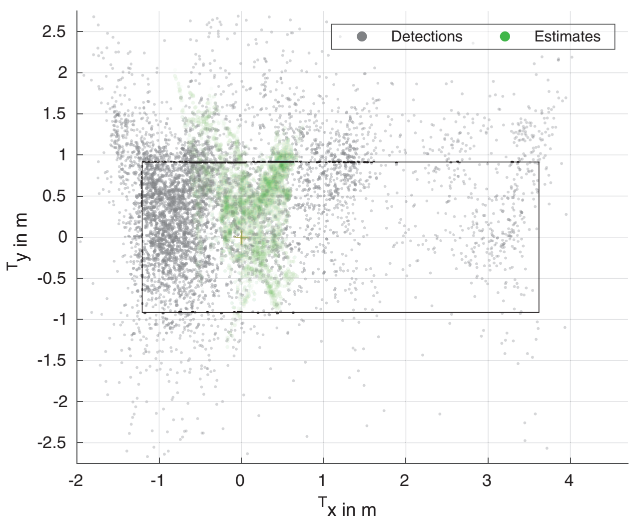


Fig. 11. Country road trailing: Position estimates in target coordinates (accumulation over whole dataset). The rectangle resembles the true extent of the vehicle.

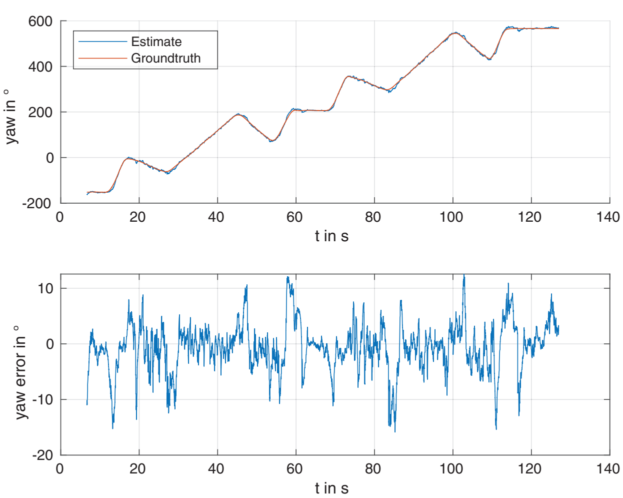


Fig. 12. Country road trailing: Yaw estimate.

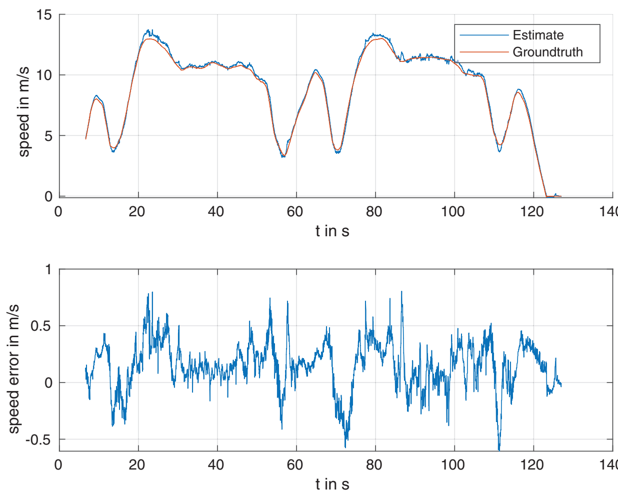


Fig. 13. Country road trailing: Speed estimate.

is subject to higher noise as they are the highest-order states of the CTRV model. Figure 15 shows the length estimation. After the first curve and progressing kinematic estimation, it is steadily improving. An error of roughly 5% remains. The repository referred in Section I-C contains a video that illustrates the association technique based on the first curve of this scenario.

B. Circling

In this scenario, the target is a long-wheelbase luxury sedan. Its extent exceeds the dimensions of the vehicles used as reference in the modeling. Besides, the radar sensors of the ego vehicle are slightly more recent and provide more detections but also more clutter measurements than the ones used in the first scenario. In addition, two additional sensors are mounted near the centers of the left and right vehicle side. The target vehicle drives circles around the parked ego vehicle, and is thus perceived by all sensors in rotating manner. However, only one side of the target vehicle is observed. Al-

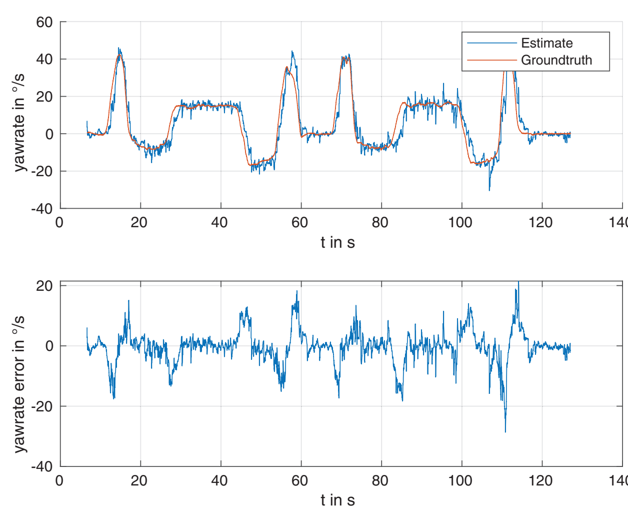


Fig. 14. Country road trailing: Yaw rate estimate.

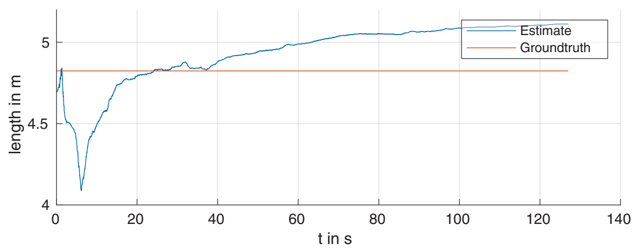


Fig. 15. Country road trailing: Length estimate.

though the lateral measurement noise is lower than in the previous example, the lateral association problem arises for the entire vehicle side. The longitudinal movement of the car is mainly inferred by detections at the ends of the side because they restrict the possible longitudinal position. In this scenario, the tracking algorithm benefits from the different component characteristics for the association and subsequent position estimation. Figures 16 and 17 show the position estimates, while Figs. 18–20 outline the estimation of the yaw, the speed and the yaw rate over time. Figure 21 shows the length estimation. It has settled from the 20th second. This scenario shows an interesting effect. Although the sensors only observe the left side of the target, a width estimation is feasible if the initial extent uncertainty is chosen accordingly: The filter inherently exploits the visibility of the opposite wheels to directly infer the width state. Figure 22 shows its estimation over time, although the resulting accuracy is attributable to the long observation time. This specific figure has been obtained by utilizing an initial width uncertainty of $\sigma_{\mathrm{w},0}^2 = 0.2\ \mathrm{m}^2$. This parametrization, though, is far higher than the statistical variety of typical vehicles and needlessly reduces the robustness of the filter especially in challenging situations.

C. Urban Trailing

The ego vehicle follows the target vehicle again in this scenario. The target vehicle and the sensors are identical to the those utilized in the second scenario. Nearby metallic containers and buildings on this narrow track cause mirrored (ghost) detections and signal overexposures. They lead to biased detections and the loss of detections from the target vehicle. The estimation performance decreases, especially concerning the yaw due to the biased detections, but the tracking stays robust. Figures 23–27 show the respective tracking performance. Figure 28 illustrates the length estimate. Again, after the kinematic quantities have roughly been estimated, it is able to resolve the initial extent error. It is steadily improving as the vehicle is mostly observed from behind.

Table III depicts a root-mean-square error (RMSE) comparison of all scenarios. Changes of the parametrization of the CTRV process noise in the range of ±20% (standard deviation) have not shown a worse degradation than 6% of the RMSE of any state variable; some state variables also show better accuracy. The position of features like the wheels is yet purely statistically derived, and its modeling error and the extent are mod-

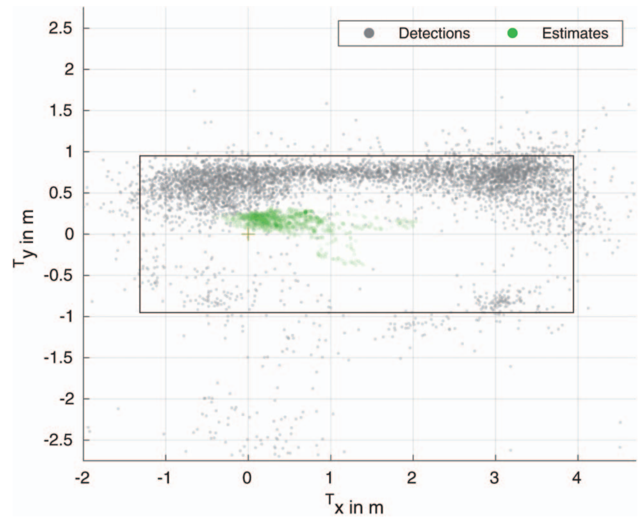


Fig. 17. Circling around the ego: Position estimates in target coordinates.

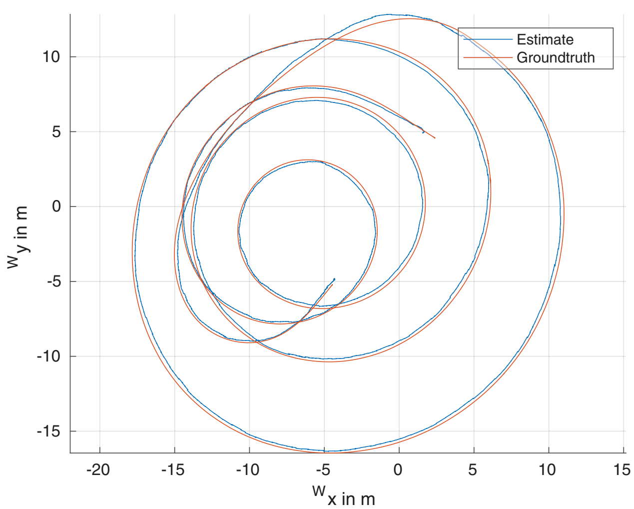


Fig. 16. Circling around the ego: Position estimate in world coordinates.

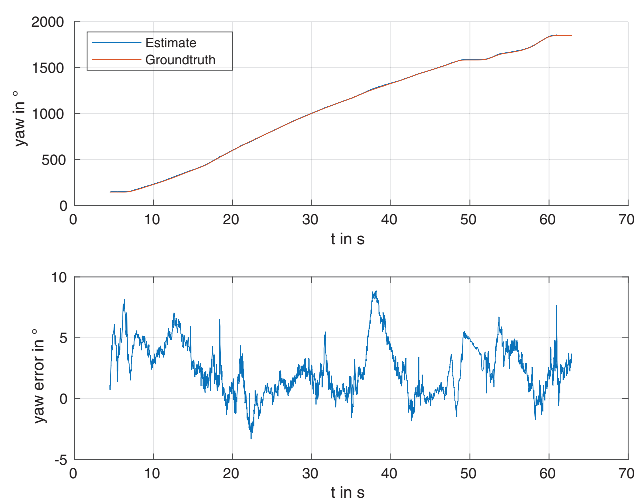


Fig. 18. Circling around the ego: Yaw estimate.

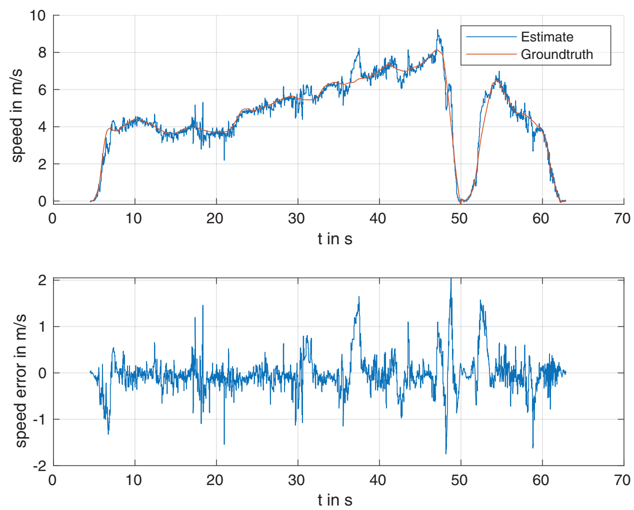


Fig. 19. Circling around the ego: Speed estimate.

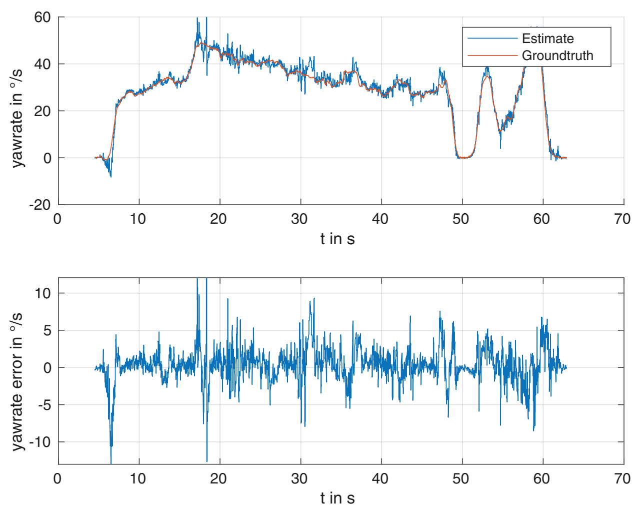


Fig. 20. Circling around the ego: Yawrate estimate.

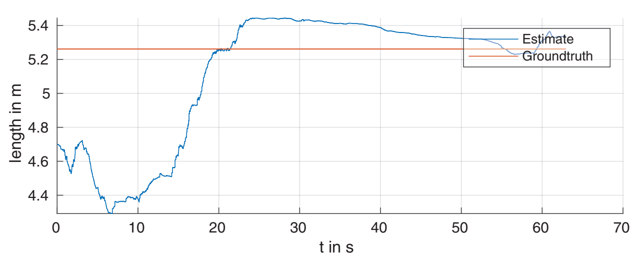


Fig. 21. Circling around the ego: Length estimate.

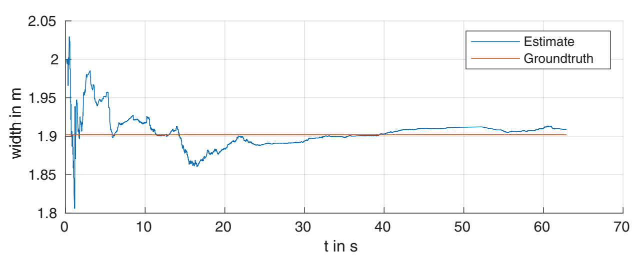


Fig. 22. Circling around the ego: Width estimate.

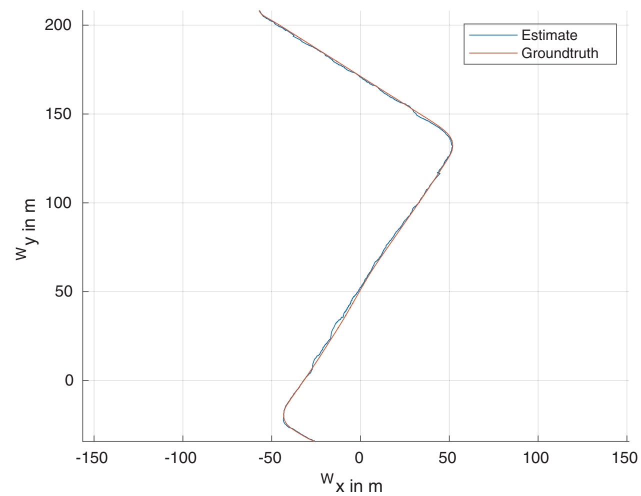


Fig. 23. Urban trailing: Position estimate in world coordinates.

Table III
Tracking Performance

RMSE	${}^{T}x$	${}^{T}y$	${}^{W}\varphi$	${}^{W}v$	${}^{W}\omega$
1) Country road	0.34 m	0.66 m	4.3°	0.25 ms^{-1}	5.4°s^{-1}
2) Circling	0.60 m	0.18 m	3.2°	0.39 ms^{-1}	2.3°s^{-1}
3) Urban trailing	0.30 m	0.69 m	5.1°	0.15 ms^{-1}	4.3°s^{-1}

The Position Error References the CTRV Pivot Point and is Given in Target Coordinates T, While the Other Errors are Given in World Coordinates W.

eled with a Gaussian noise term. A possible improvement is to correlate this noise term with the estimated extent size. However, the induced change is negligible considering typical vehicles. The incorporation of the position of such features in the state vector and their explicit estimation support the extent estimation, as the individual modeling error can be corrected over time. However, their precise estimation requires an observation time that exceeds typical urban scenarios (consider-

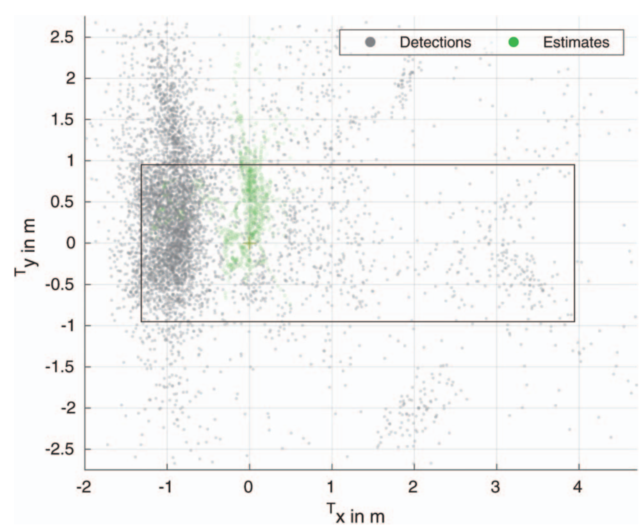


Fig. 24. Urban trailing: Position estimate in target coordinates.

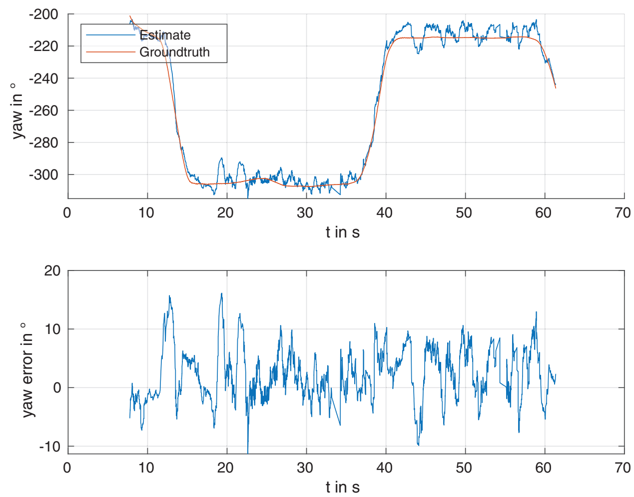


Fig. 25. Urban trailing: Yaw estimate.

ing sparse measurements). Next to groundtruth-assisted scenarios, our dataset also contains urban scenarios with numerous vehicles. As no ground truth is available for those vehicles, only a qualitative evaluation of the robustness using LiDAR scans could be performed there.

IX. TRACKING FRAMEWORK INTERFACES

This section outlines interfaces which integrate the proposed filter into larger tracking frameworks.

A. Interfaces to Low-Level Fusion Algorithms

The abstraction of an object to its physical components offers a convenient opportunity to fuse heterogeneous sensor data. Camera sensors and their processing chains often utilize semantic segmentation to detect features of objects. Following the example of vehicles, these are wheels, lights, license plates, and corners. Furthermore, LiDAR sensors detect license plates particularly well due to their reflectivity. If mounted closer to the ground, they also obtain point measurements from opposite wheels.

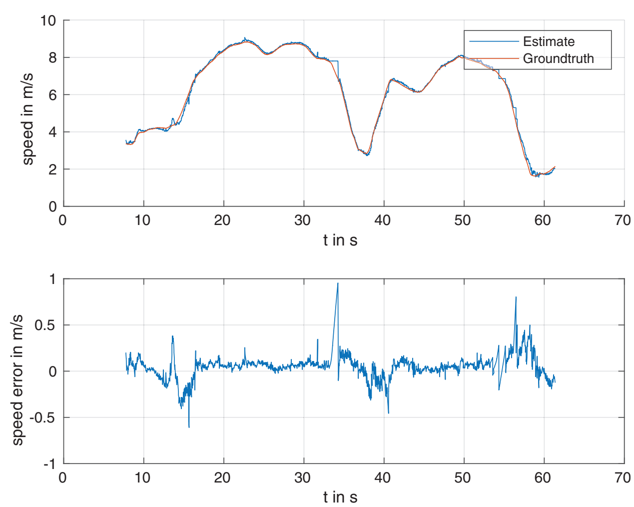


Fig. 26. Urban trailing: Speed estimate.

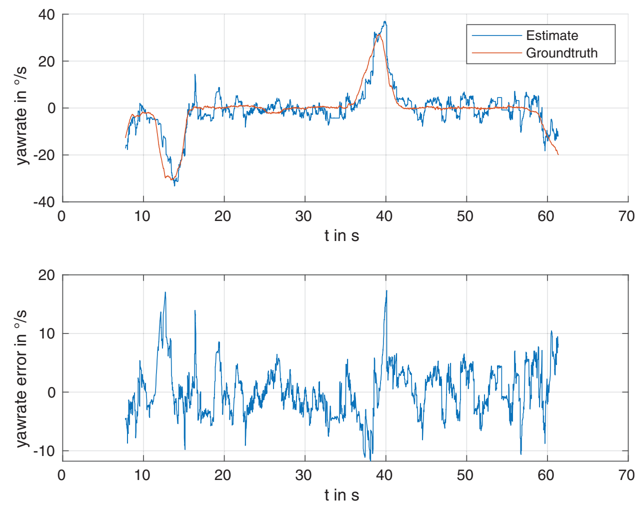


Fig. 27. Urban trailing: Yaw rate estimate.

Those features are either already directly observable with the radar sensor (e.g., wheels and corners) or can be added as additional components. The description of additional components with the parameter set A correlates them to the state vector. The abstraction of objects into components is thus a suitable interface for a low-level or feature-level fusion.

B. Interfaces to Multi-Object Trackers

We use this radar tracker in interaction with a multi-hypotheses track management in a C++/ROS-based real-time tracking application. Although the tracking shows robustness against local optima like turned vehicles or wrong wheel associations, a multihypotheses tracker speeds up the correction. The track management usually requires some additional interfaces to the underlying trackers in addition to the actual state updates of the hypotheses. Due to the size of this article, an actual implementation of a multi-target tracking and the interaction of objects cannot be covered here.

1) Track Initialization: When a track for a new object is created, either because it enters the range of visibility or it leaves an occluded area, the first obtained measurement is usually a single detection y. The center of the new object is normally set to the position of the detection as

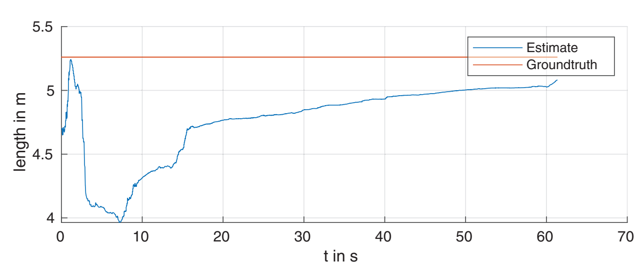


Fig. 28. Urban trailing: Length estimate.

its orientation, and therefore its side facing the sensor, is unknown. Instead, we exploit the radial speed measurement to obtain a first rough velocity vector $\mathbf{v}_W$ in world coordinates. Subsequently, we derive the orientation of the object φ (by assuming that it moves forward), determine the facing side, and place this side on the obtained detection. The velocity vector $\mathbf{v}_W$ is the sum of the velocity vector of the radar sensor $\mathbf{v}_S$ (see equation (15)) and the rotated radial speed measurement $\dot{r}$, which is transformed to world coordinates:

$$\mathbf{v}_W = \mathbf{v}_S + \mathrm{R}\left(\Theta(\mathbf{y})\right) \cdot [\dot{r}, 0]^\intercal, \tag{63}$$

and results in the initial yaw φ and speed v estimate:

$$\varphi = \operatorname{atan2}(\mathbf{v}_W), \tag{64}$$

$$v = \|\mathbf{v}_W\|. \tag{65}$$

The orientation to the detection originating from the sensor $\Theta(\mathbf{y})$ is provided by equation (17). The radar-facing side of the object is now determined and placed in the position of the detection. If no additional information on the lateral position is available, then the center of the side can be simply placed on the detection position. Although only one component of the velocity vector of the target can be measured, this approach reduces the initialization time significantly. If further detections are measured, then a complete velocity vector can be obtained [22].

We also use the radial speed measurement to determine if an unassociated detection belongs to a dynamic object. If the longitudinal component of $\mathbf{v}_S$, regarded in the sensor frame S, plus the radial speed measurement $\dot{r}$ is above a noise-dependent threshold, a new object hypothesis is created.

2) Expected Number of Detections: Track existence checks require the expected number of detections N that an object presumably generates. This number corresponds to the sum of the detection occurrence likelihoods of all components:

$$N = \sum_{j \in J} \mathrm{o}^{(j)}. \tag{66}$$

3) Detection-to-Track Association Likelihood: The crucial problem multi-object trackers deal with is the association of a detection to multiple plausible objects. This process demands the association likelihoods of one detection to all of these objects. The association likelihood for a complete object γ and a given detection measurement $\mathbf{y}$ is the sum of the association likelihoods of all components for this detection:

$$\gamma(\mathbf{y}) = \sum_{j \in J} \gamma^{(j)}(\mathbf{y}). \tag{67}$$

The radial speed measurement appears here as a valuable support for the detection-to-track association.

X. DISCUSSION

The focus of this work is the both precise and fast filtering of sparse radar detections. The problem is split into two parts: an accurate modeling and prediction of measurement data, and the respective state update using this representation. The usage of a set of components to model the measurements is conformable and predicts all relevant measurement effects. The noise of the measurement data does not justify any further particularities in our case. The state update is performed with a probabilistic association approach, which shows robust results and demands far less computing resources than combinatorial approaches. The computing time is a crucial factor as the upstream track management itself usually utilizes multi-hypotheses approaches. A single Intel i7-4790k (2014) core performs a typical complete state update in MATLAB within 2 ms, and in automatically generated C within 20 µs.

However, a difficulty results from the *base point error*. Besides the measurement matrices, also the visibility computations rely on the current state estimate. Estimates always differ from the true state. Assuming an example vehicle that is observed from the front, slight changes in the yaw estimate predict either a good visibility of the left or the right corner of the vehicle (similar to the actual physical characteristic). A workaround is to exploit multiple samplings and evaluations of the measurement functions in relation to the uncertainty of the estimate. However, this effect plays a minor role and workarounds are not necessarily required according to our findings: If a component is wrongly assumed to have just become visible, then its estimated angular extent is still small and invokes a weak, thus insignificant detection likelihood.

A major benefit is the inherent dealing with the *error in variables*. The filter considers all the uncertainties of both the state and measurements, as well as all possible sources of origin. This is in contrast to models, which assume that the regression variables can be determined in an exact manner. For example, most radial functions determine and apply the difference between measurement and contour solely in radial direction and ignore any measurement errors in the tangential direction. Dedicated precautions [26], [40] have to be applied to circumvent this issue.

To use this approach for different objects, component descriptions need to be modeled and parametrized statistically. Machine learning approaches, which derive the object characteristics from measurement data, can be deployed more quickly and are easier to implement. On the other hand, our proposed algorithm offers a low run time. Its parameterability allows the exchange of sensors without extensively recording new data. As the algorithm is based on physical relations, the model is scalable for any desired precision. The behaviour of the algorithm is deterministic and predictable. It can be modularized and extended with interfaces for a low-level fusion.

XI. CONCLUSION

This work deals with the filtering of sparse measurement data. When techniques like feature recognition in measurement data are no longer conceivable, the measurements need to be associated with their estimated origins that are derived from the state vector. We split the object into a set of different components with different measurement characteristics to compute the association more robustly. Association uncertainties are resolved probabilistically to achieve a low computing expense. This approach shows remarkable results in the prediction of measurements according to real-data comparisons. It also places a low demand on computing resources. A tracking evaluation proves the possibility of robust tracking with a low number of measurements. This low number of detections would usually not allow an accurate object state estimation with a single measurement epoch or a even few of them—the estimation is achieved by the filtering over time. The proposed algorithm is developed to be utilized in interaction with a multihypotheses track management and a heterogeneous low-level sensor fusion.

XII. FUTURE WORK

This article outlines the usage of the probabilistic component association with sparse radar measurement data and applies it for vehicle tracking. Further works can focus on other objects like pedestrians, cyclists, and trucks. After an evaluation of the usage of camera and LiDAR sensors, the algorithm can be extended with heterogeneous sensor data fusion. With the availability of radar sensors with an adequate elevation measurement performance, the component descriptions can be extended to 3D models.

The extension of the PDAF adaption to a joint probabilistic data association filter (JPDAF) variant might further improve the performance when obtaining dense measurement data [17].

A statistical study on the structure of objects and/or their measurement characteristics based on a large sample with spatial and kinematic reference data can improve the generality of the component descriptions.

Our current work, though, continues the proposed approach to support (and exploit) mutual occlusion.

ACKNOWLEDGMENT

The authors would like to thank Dominik Kellner, Andreas Rauch, Dominik Bauch, Marco Baumgartl, and colleagues at the BMW group for their support and constructive collaboration.

REFERENCES

[1] Y. Bar-Shalom, F. Daum, and J. Huang
"The probabilistic data association filter,"
IEEE Control Syst. Mag., vol. 29, no. 6, pp. 82–100, Dec. 2009.

[2] P. Berthold, M. Michaelis, T. Luettel, D. Meissner, and H.-J. Wuensche
"Radar reflection characteristics of vehicles for contour and feature estimation,"
in *Proc. Sensor Data Fusion: Trends, Solutions, Appl.*, 2017, pp. 1–6.

[3] P. Berthold, M. Michaelis, T. Luettel, D. Meissner, and H.-J. Wuensche
"An abstracted radar measurement model for extended object tracking,"
in *Proc. IEEE Intell. Transp. Syst. Conf.*, 2018, pp. 770–76.

[4] P. Berthold, M. Michaelis, T. Luettel, D. Meissner, and H.-J. Wuensche
"A radar measurement model for extended object tracking in dynamic scenarios,"
in *Proc. IEEE Intell. Veh. Symp.*, 2019, pp. 770–776.

[5] P. Berthold, M. Michaelis, T. Luettel, D. Meissner, and H.-J. Wuensche
"A continuous probabilistic origin association filter for extended object tracking,"
in *Proc. IEEE Int. Conf. Multisensor Fusion and Integration*, 2020, pp. 323–329.

[6] S. Bordonaro, P. Willett, and Y. Bar-Shalom
"Consistent linear tracker with converted range, bearing, and range rate measurements,"
IEEE Trans. Aerosp. Electron. Syst., vol. 53, no. 6, pp. 3135–3149, Dec. 2017.

[7] S. Bordonaro, P. Willett, Y. Bar-Shalom, M. Baum, and T. Luginbuhl
"Extended object tracking with exploitation of range rate measurements,"
ISIF J. Adv. Inf. Fusion, vol. 12, pp. 228–242, Dec. 2017.

[8] P. Burger, B. Naujoks, and H.-J. Wuensche
"Fast dual-decomposition based mesh-graph clustering for point clouds,"
in *Proc. IEEE Intell. Transp. Syst. Conf.*, 2018, pp. 1129–1135.

[9] M. Bühren and B. Yang
"Automotive radar target list simulation based on reflection center representation of objects,"
in *Proc. Int. Workshop Intell. Transp.*, 2006, pp. 161–166.

[10] J. Ebert, T. Gumpp, S. Münzner, A. Matskevych, A. P. Condurache, and C. Gläser
"Deep radar sensor models for accurate and robust object tracking,"
in *Proc. IEEE Intell. Transp. Syst. Conf.*, 2020, pp. 1–6.

[11] B. Forkel and H.-J. Wuensche
"LiDAR-SGM: Semi-global matching on LiDAR point clouds and their cost-based fusion into stereo matching,"
in *Proc. IEEE Int. Conf. Robot. Automat.*, 2023, pp. 1–7.

[12] K. Gilholm, S. Godsill, S. Maskell, and D. Salmond
"Poisson models for extended target and group tracking,"
Proc. SPIE, vol. 5913, pp. 230–241, Sep. 2005.

[13] K. Gilholm and D. Salmond
"Spatial distribution model for tracking extended objects,"
IEE Proc. Radar, Sonar Navigation, vol. 152, no. 5, pp. 364–371, Oct. 2005.

[14] K. Granström, M. Baum, and S. Reuter
"Extended object tracking: Introduction, overview and applications,"
ISIF J. Adv. Inf. Fusion, vol. 12, no. 2, pp. 139–174, Dec. 2017.

[15] D. Gubelli, O. A. Krasnov, and O. Yarovyi
"Ray-tracing simulator for radar signals propagation in radar networks,"
in *Proc. 10th Eur. Radar Conf.*, 2013, pp. 73–76.

[16] L. Hammarstrand, M. Lundgren, and L. Svensson
"Adaptive radar sensor model for tracking structured extended objects,"
IEEE Trans. Aerosp. Electron. Syst., vol. 48, no. 3, pp. 1975–1995, Jul. 2012.

[17] L. Hammarstrand, L. Svensson, F. Sandblom, and J. Sorstedt "Extended object tracking using a radar resolution model," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 48, no. 3, pp. 2371–2386, Jul. 2012.

[18] P. Held, D. Steinhauser, A. Kamann, A. Koch, T. Brandmeier, and U. T. Schwarz "Micro-Doppler extraction of bicycle pedaling movements using automotive radar," in *Proc. IEEE Intell. Veh. Symp.*, 2019, pp. 744–749.

[19] J. Honer and H. Kaulbersch "Bayesian extended target tracking with automotive radar using learned spatial distribution models," in *Proc. IEEE Int. Conf. Multisensor Fusion Integration*, 2020, pp. 316–322.

[20] A. Kamann, D. Steinhauser, F. Gruson, T. Brandmeier, and U. T. Schwarz "Extended object tracking using spatially resolved micro-Doppler signatures," *IEEE Trans. Intell. Veh.*, vol. 6, no. 3, pp. 440–449, Sep. 2021.

[21] H. Kaulbersch, J. Honer, and M. Baum "EM-based extended target tracking with automotive radar using learned spatial distribution models," in *Proc. Int. Conf. Inf. Fusion*, 2019, pp. 1–8.

[22] D. Kellner, M. Barjenbruch, K. Dietmayer, J. Klappstein, and J. Dickmann "Instantaneous lateral velocity estimation of a vehicle using Doppler radar," in *Proc. Int. Conf. Inf. Fusion*, 2013, pp. 877–884.

[23] C. Knill, A. Scheel, and K. Dietmayer "A direct scattering model for tracking vehicles with high-resolution radars," in *Proc. IEEE Intell. Veh. Symp.*, 2016, pp. 298–303.

[24] J. W. Koch "Bayesian approach to extended object and cluster tracking using random matrices," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 44, no. 3, pp. 1042–1059, Jul. 2008.

[25] A. I. Leonov and K. I. Fomichev *Monopulse Radar*. Norwood, MA: Artech House, 1986.

[26] M. Michaelis, P. Berthold, T. Luettel, D. Meissner, and H.-J. Wuensche "Extended object tracking with an improved measurement-to-contour association," in *Proc. Int. Conf. Inf. Fusion*, 2020, pp. 1–6.

[27] M. Michaelis, P. Berthold, D. Meissner, and H.-J. Wuensche "Heterogeneous multi-sensor fusion for extended objects in automotive scenarios using Gaussian processes and a GMPHD-filter," in *Proc. Sensor Data Fusion: Trends, Solutions, Appl.*, 2017, pp. 1–6.

[28] B. Naujoks, P. Burger, and H.-J. Wuensche "The greedy dirichlet process filter—An online clustering multi-target tracker," in *Proc. IEEE Glob. Conf. Signal Inf. Process.*, 2018, pp. 1233–1237.

[29] H. Rohling "Radar CFAR thresholding in clutter and multiple target situations," *IEEE Trans. Aerosp. Electron. Syst., AES*, vol. 19, no. 4, pp. 608–621, Aug. 1983.

[30] A. Scheel and K. Dietmayer "Tracking multiple vehicles using a variational radar model," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 10, pp. 3721–3736, Oct. 2019.

[31] R. Schubert, E. Richter, and G. Wanielik "Comparison and evaluation of advanced motion models for vehicle tracking," in *Proc. Int. Conf. Inf. Fusion*, 2008, pp. 1–6.

[32] D. Steinhauser, P. Held, A. Kamann, A. Koch, and T. Brandmeier "Micro-Doppler extraction of pedestrian limbs for high resolution automotive radar," in *Proc. IEEE Intell. Veh. Symp.*, 2019, pp. 764–769.

[33] R. L. Streit and T. E. Luginbuhl "Probabilistic multi-hypothesis tracking," Naval Undersea Warfare Center Division, Newport, RI, USA, Tech. Rep. ADA298501, Feb. 1995.

[34] D. Svensson "Derivation of the discrete-time constant turn rate and acceleration motion model," in *Proc. Sensor Data Fusion: Trends, Solutions, Appl.*, 2019, pp. 1–5.

[35] N. Wahlström and E. Özkan "Extended target tracking using Gaussian processes," *IEEE Trans. Signal Process.*, vol. 63, no. 16, pp. 4165–4178, Aug. 2015.

[36] T. A. Wheeler, M. Holder, H. Winner, and M. Kochenderfer "Deep stochastic radar models," in *Proc. IEEE Intell. Veh. Symp.*, 2017, pp. 47–53.

[37] Y. Xia, P. Wang, K. Berntorp, L. Svensson, K. Granström, H. Mansour, P. Boufounos, and P. V. Orlik "Learning-based extended object tracking using hierarchical truncation measurement model with automotive radar," *IEEE J. Sel. Topics Signal Process.*, vol. 15, no. 4, pp. 1013–1029, Jun. 2021.

[38] D. Zhang and G. Lu "Study and evaluation of different fourier methods for image retrieval," *Image Vis. Comput.*, vol. 23, no. 1, pp. 33–49, Nov. 2005.

[39] L. Zhang, H. Jiang, Y. Zhou, Y. Zhai, D. Zhou, and Y. Bai "Wheeled vehicle detection under the asymmetric micro-Doppler interference," *IEEE Sensors J.*, vol. 22, no. 18, pp. 18085–18092, Sep. 2022.

[40] S. Zhou, D. Pati, T. Wang, Y. Yang, and R. J. Carroll "Gaussian processes with errors in variables: Theory and computation," *arXiv: Statistics Theory*, 2019.

[41] Y. Zhou, L. Liu, H. Zhao, M. López-Benítez, L. Yu, and Y. Yue "Towards deep radar perception for autonomous driving: Datasets, methods, and challenges," *Sensors*, vol. 22, no. 11, Art. no. 4208, 2022.

Philipp Berthold studied electrical engineering at the Technical University of Munich (TUM), München, Germany. He has actively developed autonomous systems and localization techniques. Since 2016, he has been a Research Assistant at the Institute for Autonomous Systems Technology at the University of the Bundeswehr Munich. His current research involves heterogeneous multisensor data fusion for automated driving with focus on radar perception.

Martin Michaelis studied mathematics at the University of Bonn, Bonn, Germany. Afterwards, he worked for two years as a Research Assistantt at the Fraunhofer Institute FKIE, Wachtberg, Germany. Since 2016, he has been with the University of the Bundeswehr Munich. His main research interests are multiobject tracking, multisensor fusion, automotive radar perception, and extended target tracking.

Thorsten Luettel studied electrical engineering with a focus on mechatronics at Leibniz Universität Hannover, Hannover, Germany, where he worked primarily on bipedal walking robots. Since 2006, he has been conducting research in the field of autonomous driving at the Institute for Autonomous Systems Technology at the University of the Bundeswehr Munich, Neubiberg, Germany. His main areas of interest are sensor data fusion and system integration. As a research associate, he has led several externally funded projects in recent years as well as led "Team MuCAR" to success in international competitions. Since 2020, he also leads the research area "Networking and Autonomy" in the dtec.bw-funded project MORE (Munich Mobility Research Campus).

Daniel Meissner was born in 1984 in Ansbach, Germany. He received a Diploma degree (equivalent to M.Sc. degree) in electrical engineering from the Technical University of Munich (TUM), München, Germany, in 2009, and the Dr.-Ing. degree (equivalent to Ph.D.) in 2015 from Ulm University, Ulm, Germany. In 2014, he joined the BMW Group as a developer for driver assistance systems. In 2021, he became the Chief Engineer for object and freespace detection.

Hans-Joachim Wuensche got his Ph.D. from the University of the Bundeswehr Munich, Neubiberg, Germany, in 1987 with Ernst D. Dickmanns, where he codeveloped the 4D-approach to computer vision. After many years in management, he returned to the same university to lead the Institute for Autonomous Systems Technology in 2004. His research interests include autonomous robots, especially on- and off-road vehicles exploring and navigating unknown terrain.

INTERNATIONAL SOCIETY OF INFORMATION FUSION

ISIF Website: http://www.isif.org

SOCIETY VISION

The International Society of Information Fusion (ISIF) is the premier professional society and global information resource for multidisciplinary approaches for theoretical and applied information fusion technologies.

SOCIETY MISSION

Advocate

To advance the profession of fusion technologies, propose approaches for solving real-world problems, recognize emerging technologies, and foster the transfer of information.

Serve

To serve its members and engineering, business, and scientific communities by providing high-quality information, educational products, and services.

Communicate

To create international communication forums and hold international conferences in countries that provide for interaction of members of fusion communities with each other, with those in other disciplines, and with those in industry and academia.

Educate

To promote undergraduate and graduate education related to information fusion technologies at universities around the world. Sponsor educational courses and tutorials at conferences.

Integrate

Integrate ideas from various approaches for information fusion, and look for common threads and themes– look for overall principles, rather than a multitude of point solutions. Serve as the central focus for coordinating the activities of world-wide information fusion related societies or organizations. Serve as a professional liaison to industry, academia, and government.

Disseminate

To propagate the ideas for integrated approaches to information fusion so that others can build on them in both industry and academia.

Call for Papers

The Journal of Advances in Information Fusion (JAIF) seeks original contributions in the technical areas of research related to information fusion. Authors are encouraged to submit their manuscripts for peer review http://isif.org/journal.

Call for Reviewers

The success of JAIF and its value to the research community is strongly dependent on the quality of its peer review process. Researchers in the technical areas related to information fusion are encouraged to register as a reviewer for JAIF at http://jaif.msubmit.net. Potential reviewers should notify via email the appropriate editors of their offer to serve as a reviewer.

Journal of Advances in Information Fusion Volume 17 Number 2 December 2022 ISSN 1557-6418