June 2014 Volume 9 Number 1 ISSN 1557-6418

JAIF 14

Journal of Advances in Information Fusion

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

Regular Papers Page

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.

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.

From the Editor-in-Chief:

June 2014

Past, Present, and Future of JAIF

In January 2014, I took over the role as Editor-in-Chief of this journal from William Dale Blair. I would like to take the opportunity to thank Dale for serving as Editor-in-Chief for the past eight years and for his continuing professional, knowledgeable, and patient guidance of the editorial team, the technical editors, the authors, and the staff.

The Journal of Advances in Information Fusion (JAIF) was founded in 2006 as the flagship journal of the International Society of Information Fusion (ISIF) to serve the growing demand for high-quality publications of information fusion methods. JAIF is an open-source, peer-reviewed, semi-annual archival journal published electronically and distributed via the internet.

I think it is fair to say that so far we kept our promise on delivering high-quality publications. Beginning with the inaugural issue in July 2006, 17 issues have been published up to now that are freely available at http://www.isif.org/journal.

The quality of JAIF is maintained by high standards for the peer review process through an editorial board with strong academic and industrial backgrounds. Prior to publication, each manuscript requires a review from at least three referees and manuscript corrections that address any shortcoming identified by the referees and editors. A web-based review system at http://jaif.msubmit.net is used for handling the peer review of manuscripts electronically. This system facilitates the review of manuscripts for authors and archives the reviewers' comments and editorial decisions for all manuscripts.

Journal publications nowadays play an important role when it comes to evaluating the quality of individual researchers. Bibliometric indicators are increasingly used to rate scientific publications and the researchers who authored them. This is then used by employers such as universities, companies, and government organizations to make decisions about raises or promotions or funding decisions concerning grants. Fortunately, JAIF is now indexed at SCOPUS, so that research published at JAIF becomes more visible. At this point, I would like to thank our Administrative Editor Robert Lynch for his relentless effort on getting the journal indexed, which is known to be a tedious and lengthy process.

In the future, we seek to increase the number of submissions to the journal, so that it might eventually be possible to publish four issues per year instead of two without compromising the paper quality. This will hopefully happen automatically as more and more potential authors appreciate the quality of JAIF. In addition, we will make the journal more interesting by encouraging special issue proposals.

Special issues of JAIF are intended to cover a topic of special interest in the area of information fusion. Potential guest editors should submit a formal proposal for expressing their desire to organize a special issue. Formally, these proposals should provide the biographical information of the guest editors, the motivation for the special issue, a general call for contributed papers to the special issue, a candidate list of papers with authors for direct invitation, and a schedule for delivery and review of the papers for the special issue. Typically, one or two guest editors will be responsible for the special issues, and they will be supported by a small team of predefined reviewers in order to expedite the review process.

Finally, I hope that I convinced you that JAIF is a high-quality journal dedicated to research on information fusion methods. So, if you have the feeling that your material could meet the journal's standards, please consider submitting it to JAIF.

Uwe D. Hanebeck
Editor-in-Chief

A Technique for Deriving Multitarget Intensity Filters Using Ordinary Derivatives

ROY STREIT

In multitarget tracking problems based on finite point process models of targets and measurements, it is known that the distribution of the Bayes posterior point process is a ratio of functional derivatives of a joint probability generating functional. It is shown here that these functional derivatives can be found by evaluating ordinary derivatives. The method is exact, not approximate. Several examples are presented, including multisensor target tracking and extended-target tracking. The method is well suited to the needs of particle filter implementations.

Manuscript received September 29, 2012; revised June 30, 2013 and October 26, 2013; released for publication October 29, 2013.

Refereeing of this contribution was handled by Stefano Coraluppi.

Author's addresses: Metron, Inc., 1818 Library St., Reston, VA 20109.

1557-6418/14/$17.00

1. INTRODUCTION

This paper shows that functional derivatives of the probability generating functional (PGFL) of a finite point process can be calculated using ordinary derivatives. The result is new, and it is potentially useful to the class of Bayesian multitarget tracking problems that is based on finite point process models for targets and measurements. In this class, the distribution of the Bayes posterior multitarget process is a ratio of functional derivatives of the joint measurement-target PGFL. In some problems evaluating the functional derivatives is only a tedious task, but in other problems the number of terms in the derivatives is prohibitively large and limits practical applications of the method.

The proof is straightforward—we reduce the PGFL to an ordinary function that is conceptually straightforward to differentiate. This function is called a secular function to emphasize that it is an "ordinary" function and not a functional. Existing symbolic software packages can be used to differentiate the secular function, a fact that is potentially of practical importance since software for functional differentiation of the PGFL does not seem to be available. The methods of this paper use the established theory of PGFLs and their functional derivatives.

The proposed methods are compatible with particle, or sequential Monte Carlo (SMC), filter implementations. The basic strategy is to embed symbolic differentiation software in the production code and evaluate the symbolic derivatives of the secular functions at the points of the particle filter. One of the purposes of this paper is to show that this is a theoretically feasible strategy. Its practical utility is outside the scope of the paper.

Two tracking applications where functional differentiation causes serious difficulties are discussed. One is multisensor target tracking [9]. The other is extended-target tracking problems in which targets can produce more than one measurement [8, 11]. The secular functions for both problems are derived.

Functional differentiation of the PGFL is the result of a double limit. A theoretical question naturally arises, "Can these limits be interchanged?" The answer is, "Yes, for the problems of interest here." This result seems to be new. It gives a better understanding of PGFLs and their relationship to classical probability generating functions (PGFs).

Section II speaks of the PGFL as an encoding of the multitarget tracking problem and functional differentiation of the PGFL as the decoding algorithm. Section III gives a simple example of the method we use to reduce PGFLs to secular functions. Section IV proves that for the class of PGFLs of interest in this paper, ordinary derivatives of secular functions are *identical* to functional derivatives of PGFLs. Section V gives several examples of secular functions, including those for multisensor and extended-target tracking problems. Section VI discusses finite differences and series expansion

methods for approximating secular functions. Section VII gives conclusions.

2. FINITE POINT PROCESSES AND PGFLS

The grand canonical ensemble is defined to be the set of all finite lists of points in a given space. A random variable whose outcomes are in this ensemble is called a finite point process. When the space is continuous (i.e., it has no discrete elements) and the probability distribution is orderly (i.e., the underlying Borel measure has no atoms), realizations of the finite point process cannot have repeated elements, that is, the outcomes of the finite point process are sets with probability one.

Some familiarity is assumed with the definitions and properties of finite point processes such as given in [1, 3, 10, 12]. Such familiarity should include PGFLs, which are defined in [10] as an expectation of a random product with respect to a probability distribution over the grand canonical ensemble. Finally, readers are assumed to be familiar with the application of PGFLs to multitarget tracking, for which see [7] or the comparison paper [13].

In applications of finite point processes to Bayesian multitarget tracking, the joint measurement-target PGFL "encodes" the over-all probability structure of the problem. The measurement-to-target/clutter assignments are assumed unknown in this paper, so the probabilistic structure is inherently combinatorial. The PGFL of the Bayes posterior process is derived from the joint PGFL by conditioning on the measurement set.

To find the Bayes posterior probability of a given event, it is necessary to "decode" its PGFL. Decoding is equivalent to functional differentiation of the PGFL. Differentiation is straightforward in principle, but it often has high computational complexity because large numbers of different kinds of terms can appear in the functional derivatives. Decoding the PGFL of the Bayes posterior point process is the problem of interest in this paper.

The primary purpose of the paper is to show that functional derivatives of the PGFL are equivalent to ordinary derivatives of functions that are easily derived from the PGFL. These functions are of independent interest, so we refer to them as secular functions. Several examples are given in Section V. The number of terms in their derivatives is prohibitively large, so the symbolic derivatives are not shown; however, the derivatives can be found using widely available software. Many software packages can be configured to evaluate the symbolic derivative numerically at specified points—thus avoiding the need for manually recoding. The secular method is exact, not approximate.

For particle tracking filter implementations, we need to evaluate the functional derivative of the PGFL for every particle (point) in the current particle set. These numerical values are used to update the particle weights, and they are subsequently used to resample the particles in the SMC update step. The same particle weights can also be computed by evaluating the symbolic derivatives of secular functions. This procedure is potentially important for applications in which derivatives are too difficult to find by hand; however, further discussion is outside the scope of the paper.

3. ILLUSTRATIVE EXAMPLE

Secular functions are obtained from PGFLs by a straightforward procedure. In this section we illustrate the technique with an example not unlike what is encountered in tracking applications.

The PGF of a Poisson distributed random number $N \geq 0$ with mean $\lambda \geq 0$ is given by $F(s) = \exp(-\lambda + \lambda s)$, where s is a complex-valued variable. Define the functional $I[g] = \int_Y g(y)q(y)dy$, where Y is a closed and bounded subset of $\mathbb{R}^2$ and $q(\cdot)$ is a continuously differentiable probability density function (PDF) in the interior of Y. The function $g(\cdot)$ is assumed to be such that $|g(y)| \leq 1$ for all $y \in Y$ and infinitely differentiable in the interior of Y. Define the functional

$$\Psi[g] = F(I[g]) = \exp\left(-\lambda + \lambda \int_Y g(y)q(y)dy\right). \quad (1)$$

Note that $\Psi[g]$ is the PGFL of the nonhomogeneous Poisson point process on Y whose intensity function is $\lambda q(\cdot)$. Using the Calculus of Variations, let the variation $\gamma(\cdot)$ be a bounded function on Y and infinitely differentiable in the interior of Y. The functional derivative of $\Psi[g]$ with respect to the variation $\gamma(\cdot)$ is defined by

$$\begin{aligned}\frac{\partial \Psi}{\partial \gamma}[g] &\equiv \left.\frac{d\Psi[g + \alpha\gamma]}{d\alpha}\right|_{\alpha=0} \\ &= \lambda\left(\int_Y \gamma(y)q(y)dy\right)\exp\left(-\lambda + \lambda\int_Y g(y)q(y)dy\right).\end{aligned} \quad (2)$$

Ordinary derivatives are denoted by "d" to distinguish them from functional derivatives which are denoted by "∂." Let c be an interior point of Y. Denote the Dirac delta function at c by $\delta_c(y)$. We will often refer to $\delta_c(y)$ as an "impulse" at the point c. Let $\gamma_n^c(y)$, $n = 1, 2, \ldots$, denote a sequence of test functions for $\delta_c(y)$. (General discussions of test functions are widely available; see the classic text [6].) There are many possible choices for $\gamma_n^c(y)$. To be specific, we take $\gamma_n^c(y)$ to be the PDF of a bivariate Gaussian random variable with mean c and covariance matrix equal to the identity matrix scaled by n^{-2} that is truncated and normalized to integrate to one on Y. Thus, $\gamma_n^c(y)$ is non-negative, infinitely differentiable interior to Y, and is unimodal with a maximum value occurring at the point c. The sequence itself is not bounded. For such test sequences it is easy to prove that

$$\lim_{n\to\infty} \int_Y q(y)\gamma_n^c(y)dy = q(c). \quad (3)$$

Note that the integral in (3) is evaluated before taking the limit. For each n, it follows from (2) that

$$\frac{\partial \Psi}{\partial \gamma_n^c}[g] = \lambda \left(\int_Y \gamma_n^c(y) q(y) dy \right) \exp\left(-\lambda + \lambda \int_Y g(y) q(y) dy \right). \tag{4}$$

The functional derivative of the PGFL $\Psi[g]$ with respect to an impulse at $c \in Y$ is the limit of (4) as $n \to \infty$. Thus, using (3),

$$\frac{\partial \Psi}{\partial c}[g] \equiv \lim_{n\to\infty} \frac{\partial \Psi}{\partial \gamma_n^c}[g] = \lambda q(c) \exp\left(-\lambda + \lambda \int_Y g(y) q(y) dy \right). \tag{5}$$

Using (1), we define

$$\begin{aligned} J(\alpha) &= \lim_{n\to\infty} \Psi[g + \alpha\gamma_n^c] \\ &= \exp\left(-\lambda + \lambda \int_Y g(y) q(y) dy + \lambda\alpha q(c) \right). \end{aligned} \tag{6}$$

The ordinary derivative $J'(0)$ is identical to (5). In this paper we call $J(\cdot)$ a secular function.

The example shows that the functional derivative of the PGFL (1) at c is identical to the derivative of its secular function (6) at zero. The next section shows that the technique extends to more general problems.

4. SECULAR FUNCTIONS

Our goal is to present results of the kind needed for tracking applications, not to give a general mathematical treatment. We assume that $F(s) = \sum_{n=0}^{\infty} \Pr\{n\} s^n$ is the PGF of a discrete random variable with outcomes in the non-negative integers, $\mathbb{N}$. Thus, $F(s)$ is analytic at the origin in the complex s-plane, $\mathbb{C}$. Because $F(1) = 1$, it is analytic in a region that includes the closed unit disc. Let Y be a closed and bounded subset of the Euclidean space $\mathbb{R}^d$, $d \geq 1$. The function $q : Y \to \mathbb{R}$ is assumed to be a PDF on Y and continuously differentiable at interior points of Y except possibly for jump discontinuities of the kind that occur, e.g., by truncating a Gaussian distribution. The function $g : Y \to \mathbb{R}$ is assumed to be such that $|g(y)| \leq 1$ for all $y \in Y$. The functional $\Psi[g]$ is defined as in the example, that is,

$$\Psi[g] = F\left(\int_Y g(y) q(y) dy \right). \tag{7}$$

Let the variation $\gamma : Y \to \mathbb{R}$ be bounded, i.e., for some number B, $|\gamma(y)| \leq B < \infty$ for all $y \in Y$. Then $\Psi[g + \alpha\gamma]$, considered as a function of the complex variable α, is analytic in an open neighborhood of the origin, i.e., in the open disc $|\alpha| < r$ where r is sufficiently small. The functional derivative of $\Psi[g]$ with respect to γ is defined by

$$\frac{\partial \Psi}{\partial \gamma}[g] = \left. \frac{d\Psi[g + \alpha\gamma]}{d\alpha} \right|_{\alpha=0}. \tag{8}$$

Substituting (7) into (8) gives, by direct calculation,

$$\frac{\partial \Psi}{\partial \gamma}[g] = F^{(1)}\left(\int_Y g(y) q(y) dy \right) \int_Y q(y) \gamma(y) dy, \tag{9}$$

where $F^{(1)}(\cdot)$ denotes the ordinary first derivative of $F(\cdot)$.

We seek the functional derivative of $\Psi[g]$ with respect to an impulse at the point $c \in Y$. As in the example, let $\gamma_n^c(y)$, $n = 1, 2, \ldots$, denote a sequence of test functions for the Dirac delta function $\delta_c(y)$ at $c \in Y$. Test functions are bounded, so the functional derivative of $\Psi[g]$ with respect to a given test function is well-defined. The functional derivative of $\Psi[g]$ with respect to an impulse at c is defined by the limit

$$\begin{aligned} \frac{\partial \Psi}{\partial c}[g] &\equiv \lim_{n\to\infty} \frac{\partial \Psi}{\partial \gamma_n^c}[g] \\ &= F^{(1)}\left(\int_Y g(y) q(y) dy \right) \lim_{n\to\infty} \int_Y q(y) \gamma_n^c(y) dy \\ &= F^{(1)}\left(\int_Y h(y) q(y) dy \right) q(c). \end{aligned} \tag{10}$$

Using the same test sequence, the secular function corresponding to $\Psi[g]$ is defined by

$$\begin{aligned} J(\alpha; c) &= \lim_{n\to\infty} \Psi[g + \alpha\gamma_n^c] \\ &= \lim_{n\to\infty} F\left(\int_Y g(y) q(y) dy + \alpha \int_Y \gamma_n^c(y) q(y) dy \right) \\ &= F\left(\int_Y g(y) q(y) dy + \alpha q(c) \right). \end{aligned} \tag{11}$$

Taking the limit inside the argument of $F(\cdot)$ is justified by analyticity. Note that J depends implicitly on g. The derivative of J with respect to α evaluated at zero is

$$\frac{dJ}{d\alpha}(0; c) = \left. \frac{d}{d\alpha} J(\alpha; c) \right|_{\alpha=0} = F^{(1)}\left(\int_Y g(y) q(y) dy \right) q(c). \tag{12}$$

Comparing (12) and (10) shows that

$$\frac{\partial \Psi}{\partial c}[g] = \frac{dJ}{d\alpha}(0; c). \tag{13}$$

Thus, the functional derivative of Ψ is identical to the ordinary derivative of its secular function J.

The result (13) does not alter the theory of PGFLs and their functional derivatives. It does, however, show that functional derivatives of the PGFL can be replaced by ordinary derivatives of the secular function. This means that many functions of interest in tracking applications (e.g., intensity and pair-correlation) can be found by differentiating the secular function of the PGFL.

The functional derivative of the PGFL is a double limit. The above results show that the order in which

these limits are taken can be interchanged. Informally,

$$\begin{aligned}\lim_{n\to\infty}\lim_{\alpha\to 0}\frac{d\Psi[g+\alpha\gamma_n^c]}{d\alpha}&=\frac{\partial\Psi}{\partial c}[g]\\&=\frac{dJ}{d\alpha}(0;c)=\lim_{\alpha\to 0}\lim_{n\to\infty}\frac{d\Psi[g+\alpha\gamma_n^c]}{d\alpha}.\end{aligned}\quad(14)$$

4.1. Extensions for Multivariate PGFs and Cross-Derivatives

The basic result (13) extends to more general functions and functionals. Suppose that

$$\Psi[g]=F\left(\int_Y g(y)q_1(y)dy,\ldots,\int_Y g(y)q_k(y)dy\right),\quad(15)$$

where $F(s_1,\ldots,s_k)$, $k\geq 1$, is the multivariate PGF of a random vector of integers with outcomes in $\mathbb{N}^k$. Hence, $F(\cdot)$ is analytic at the origin $\mathbf{0}=(0,\ldots,0)\in\mathbb{C}^k$. The functions $q_i:Y\to\mathbb{R}$, $i=1,\ldots,k$, are assumed to be continuously differentiable PDFs on Y except possibly for jump discontinuities. As before, let c be interior to Y, and let $\gamma_n^c(y)$, $n=1,2,\ldots$, be a test sequence for $\delta_c(y)$. The secular function of $\Psi[g]$ is defined to be

$$\begin{aligned}J(\alpha;c)&=\lim_{n\to\infty}\Psi[g+\alpha\gamma_n^c]\\&=\lim_{n\to\infty}F\left(\int_Y(g(y)+\alpha\gamma_n^c(y))q_1(y)dy,\ldots,\right.\\&\qquad\left.\int_Y(g(y)+\alpha\gamma_n^c(y))q_k(y)dy\right)\\&=F\left(\int_Y g(y)q_1(y)dy+\alpha q_1(c),\ldots,\right.\\&\qquad\left.\int_Y g(y)q_k(y)dy+\alpha q_k(c)\right).\end{aligned}$$

The ordinary derivative is

$$\frac{dJ}{d\alpha}(0;c)=\left.\frac{dJ}{d\alpha}\right|_{\alpha=0}=\sum_{\ell=1}^{k}F_\ell^{(1)}\left(\int_Y g(y)q_1(y)dy,\ldots,\int_Y g(y)q_k(y)dy\right)q_\ell(c),\quad(16)$$

where $F_\ell^{(1)}(\cdot)$ denotes the (ordinary) first derivative of $F(\cdot)$ with respect to argument ℓ. Direct calculation of the functional derivative of $\Psi[g]$ with respect to an impulse at c shows that it is identical to the right hand side of (16). Thus, (14) holds for PGFLs of the form (15).

The functional derivative with respect to impulses at points $\mathbf{y}=\{y_1,\ldots,y_m\}$, $m\geq 0$, is defined by

$$\Psi_{\mathbf{y}}[g]\equiv\frac{\partial^m}{\partial y_1\cdots\partial y_m}\Psi[g].\quad(17)$$

The points $y_i\in Y$ are assumed distinct. Let $\gamma_n^{y_i}(y)$, $n=1,2,\ldots$, denote a test function sequence for the Dirac delta function at y_i. The secular function of (15) is defined (cf. [10, Eqn. (4.11)] and (26)–(27) below) by substituting a test sequence for a weighted train of Dirac delta functions:

$$\begin{aligned}J(\boldsymbol{\alpha};\mathbf{y})&=\lim_{n\to\infty}\Psi\left[g(y)+\sum_{i=1}^{m}\alpha_i\gamma_n^{y_i}(y)\right]\\&=F\left(\int_Y g(y)q_1(y)dy+\sum_{i=1}^{m}\alpha_i q_1(y_i),\ldots,\right.\\&\qquad\left.\int_Y g(y)q_k(y)dy+\sum_{i=1}^{m}\alpha_i q_k(y_i)\right),\end{aligned}\quad(18)$$

where $\boldsymbol{\alpha}=(\alpha_1,\ldots,\alpha_m)^T\in\mathbb{R}^m$. The secular function is thus a function of the coefficient vector $\boldsymbol{\alpha}$ and depends implicitly on the function g. It is easily seen that

$$\Psi_{\mathbf{y}}[g]=J_{\boldsymbol{\alpha}}(\mathbf{0};\mathbf{y})\equiv\left.\frac{d^m}{d\alpha_1\cdots d\alpha_m}J(\boldsymbol{\alpha};\mathbf{y})\right|_{\alpha_1=\cdots=\alpha_m=0},\quad(19)$$

where $\mathbf{0}$ denotes the zero vector. Thus, the functional derivative of Ψ with respect to impulses at the points $\mathbf{y}\equiv\{y_1,\ldots,y_m\}$ is identical to the first order mixed derivative of the secular function. Such derivatives are called cross-derivatives in the automatic differentiation literature [5]. See Section VI for further comment on this topic.

4.2. Secular Functions for Multivariate PGFLs

Joint PGFLs correspond to two or more finite point processes defined on possibly different spaces. In tracking applications, for example, the joint PGFL can be the joint measurement-target process on the measurement space Y and the target space S. The discussion here is limited to these processes. The PGFL $\Psi[g,h]$ is assumed known. The extension to more than two processes is straightforward.

Conditioned on the (distinct) measurements $\mathbf{y}=\{y_1,\ldots,y_m\}$, $m\geq 0$, the PGFL of the Bayes posterior point process is the normalized functional derivative:

$$\Psi[h\mid\mathbf{y}]\equiv\frac{\Psi_{\mathbf{y}}[g,h]|_{g(\cdot)=0}}{\Psi_{\mathbf{y}}[g,1]|_{g(\cdot)=0}}=\frac{\Psi_{\mathbf{y}}[0,h]}{\Psi_{\mathbf{y}}[0,1]},\quad(20)$$

where the functional derivative with respect to impulses at the points of $\mathbf{y}$ is

$$\Psi_{\mathbf{y}}[g,h]\equiv\frac{\partial^m}{\partial y_1\cdots\partial y_m}\Psi[g,h].\quad(21)$$

As a check, note that $\Psi[1\mid\mathbf{y}]=1$ for all $\mathbf{y}$. The nth factorial moment of (20) is defined to be the functional derivative with respect to impulses at the (distinct) points $\mathbf{x}=\{x_1,\ldots,x_n\}$, $n\geq 0$; explicitly,

$$\begin{aligned}m_{[n]}(x_1,\ldots,x_n)&=\Psi_{\mathbf{x}}[h\mid\mathbf{y}]|_{h(\cdot)=1}\equiv\frac{\partial^n}{\partial x_1\cdots\partial x_n}\Psi[1\mid\mathbf{y}]\\&=\frac{\Psi_{\mathbf{yx}}[0,1]}{\Psi_{\mathbf{y}}[0,1]},\end{aligned}\quad(22)$$

where

$$\Psi_{\mathbf{yx}}[g,h] \equiv \frac{\partial^m}{\partial y_1 \cdots \partial y_m} \frac{\partial^n}{\partial x_1 \cdots \partial x_n} \Psi[g,h]. \tag{23}$$

Note that the nth factorial moment is a function of the point $\mathbf{x}$. For a careful definition of factorial moments, see [10], the definitive text [3], or the recent paper [13].

The first factorial moment is commonly known as the intensity function of the point process. In tracking applications it is sometimes called the probability hypothesis density (PHD). For $n = 1$, (22) can be written in an interesting logarithmic form as

$$m_{[1]}(x_1) = \frac{\partial}{\partial x_1} \log \Psi[h \mid \mathbf{y}] \bigg|_{h(\cdot)=1}. \tag{24}$$

Intuitively, the second factorial moment is a "two point" intensity function. The pair correlation function for $x_1 \neq x_2$ is defined as the ratio

$$\rho(x_1,x_2) = \frac{m_{[2]}(x_1,x_2)}{m_{[1]}(x_1) m_{[1]}(x_2)}. \tag{25}$$

From the independent sampling property of PPPs it can be shown that $m_{[2]}(x_1,x_2) = m_{[1]}(x_1) m_{[1]}(x_2)$, so that $\rho(x_1,x_2) = 1$. A point process is said to be attractive if $\rho(x_1,x_2) > 1$ for all distinct points, and repulsive if $\rho(x_1,x_2) < 1$.

The derivative of the PGFL is evaluated for the constant functions $g(\cdot) = 0$ and $h(\cdot) = 1$ to find, respectively, the event probabilities and factorial moments. For this reason we employ the simultaneous perturbations (see [10, Eqn. (4.11)])

$$g(y) = \sum_{i=1}^{m} \alpha_i \delta_{y_i}(y), \quad y \in Y, \tag{26}$$

$$h(s) = 1 + \sum_{j=1}^{n} \beta_j \delta_{x_j}(s), \quad s \in S, \tag{27}$$

where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_m) \in \mathbb{R}^m$ and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_n) \in \mathbb{R}^n$. The sums are defined to be zero for $m = 0$ and $n = 0$. Therefore, the secular function corresponding to the joint PGFL is

$$J(\boldsymbol{\alpha},\boldsymbol{\beta};\mathbf{y},\mathbf{x}) = \Psi\left[\sum_{i=1}^{m} \alpha_i \delta_{y_i}(y), 1 + \sum_{j=1}^{n} \beta_j \delta_{x_j}(s)\right]. \tag{28}$$

The test function sequence forms of (26)–(28) are somewhat tedious, so we do not use them. (The result is unchanged by using test sequence versions of these expressions.) The methods of the previous subsection show that the ordinary derivatives of J are identical to the functional derivatives of Ψ; explicitly,

$$J_{\boldsymbol{\alpha}\boldsymbol{\beta}}(\boldsymbol{\alpha},\boldsymbol{\beta};\mathbf{y},\mathbf{x})|_{\boldsymbol{\alpha}=\mathbf{0},\boldsymbol{\beta}=\mathbf{0}} \equiv \Psi_{\mathbf{yx}}[g,h]|_{g(\cdot)=0,h(\cdot)=1}. \tag{29}$$

Therefore, functional derivatives of the joint PGFL can be replaced wherever they occur by ordinary derivatives of its secular function.

Of particular interest is the first factorial moment, or intensity, of the Bayes posterior point process. This is the special case $n = 1$ of (24). Written in terms of the secular function, with $\boldsymbol{\beta} = \beta_1$ and $\mathbf{x} = x_1$,

$$m_{[1]}(x_1) = \frac{d}{d\beta_1} \log J_{\boldsymbol{\alpha}}(\mathbf{0},\beta_1;\mathbf{y},x_1) \bigg|_{\beta_1=0}. \tag{30}$$

Recall that we evaluate the derivative at $\beta_1 = 0$ because (27) is a perturbation about $h(s) = 1$. The second factorial moment of the Bayes posterior process is, from (22) with $n = 2$,

$$m_{[2]}(x_1,x_2) \equiv \frac{1}{J_{\boldsymbol{\alpha}}(\mathbf{0},\mathbf{0};\mathbf{y},\mathbf{x})} \frac{d^2}{d\beta_1 d\beta_2} J_{\boldsymbol{\alpha}}(\mathbf{0},\boldsymbol{\beta};\mathbf{y},\mathbf{x}) \bigg|_{\beta_1=\beta_2=0}, \tag{31}$$

where $\boldsymbol{\beta} = (\beta_1,\beta_2)$ and $\mathbf{x} = (x_1,x_2)$.

5. EXAMPLES OF SECULAR FUNCTIONS

Example 1 starts with the joint PGFL of the single sensor point target problem. The PHD intensity filter is derived from the secular function of the PGFL of the Bayes posterior point process. Time indexing in this and the other examples is suppressed to simplify notation. The pair-correlation function is derived in Example 2. Secular functions have little to offer in these examples because functional derivatives can be evaluated by hand. The joint PGFLs for multisensor target tracking and extended-target tracking are given in Examples 3 and 4, respectively. The corresponding secular functions are then derived. The functional derivatives cannot be found by hand for these examples but the ordinary derivatives of the secular functions can be found using reliable and efficient software. These expressions are identical to the functional derivatives of the PGFL.

5.1. Example 1: Secular Functions for the PHD/Intensity Filter

The Bayes posterior point process is not a Poisson point process (PPP). To close the Bayesian recursion, the posterior process is approximated [7] by a PPP whose intensity function is matched to the intensity, or first moment, of the Bayes posterior process. Predicting the intensity of this PPP approximation forward to the current time gives a PPP with intensity $f(s)$. After the prediction step, the joint PGFL of the measurement-target process is

$$\begin{aligned}\Psi[g,h] = \exp\Bigg[&-\int_Y \lambda(y)dy + \int_Y g(y)\lambda(y)dy - \int_S f(s)ds \\ &+ \int_S h(s)f(s)ds - \int_S h(s)P^D(s)f(s)ds \\ &+ \int_S \int_Y g(y)h(s)p(y \mid s)P^D(s)f(s)dy\,ds\Bigg],\end{aligned} \tag{32}$$

where Y and S are the sensor measurement and target state spaces, respectively, $\lambda(y)$ is the intensity function of a PPP model for sensor clutter measurements, $P^D(s)$ is the probability of detecting a target in state s, and $p(y \mid s)$ is the sensor measurement likelihood function. The PGFL is defined for bounded functions $g : Y \to \mathbb{R}$ and $h : S \to \mathbb{R}$. The PGFL (32) depends on many assumptions about target motion, target measurement, the clutter process, and the measurement-to-target assignments, to name only a few. These assumptions and the derivation of the PGFL are not given here because we take the joint PGFL as our starting point. Further details can be found in [7] and also in [13].

The secular function is defined by (28). In the case of the PGFL (32), this gives

$$\begin{aligned} J(\boldsymbol{\alpha},\boldsymbol{\beta};\mathbf{y},\mathbf{x}) &= G_0 \exp\left[\sum_{i=1}^{m}\alpha_i\left(\lambda(y_i)+\int_S p(y_i \mid s)P^D(s)f(s)ds\right)\right. \\ &\quad + \sum_{j=1}^{n}\beta_j(1-P^D(x_j))f(x_j) \\ &\quad \left. + \sum_{i=1}^{m}\sum_{j=1}^{n}\alpha_i\beta_j p(y_i \mid x_j)P^D(x_j)f(x_j)\right], \end{aligned} \tag{33}$$

where

$$G_0 = \exp\left[-\int_Y \lambda(y)dy - \int_S f(s)P^D(s)ds\right]. \tag{34}$$

Note that $J(\cdot)$ is the exponential of a quadratic polynomial in the components of the vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. The derivatives of $J(\cdot)$ are straightforward to compute by hand when $\min\{m,n\}$ is small.

The PGFL of the Bayes posterior point process is, using (20),

$$\Psi[h] = \frac{\Psi_{\mathbf{y}}[0,h]}{\Psi_{\mathbf{y}}[0,1]}. \tag{35}$$

The intensity function of this process is $m_{[1]}(x_1)$. It is given in terms of the secular function by (30), where the derivative $J_\alpha(\cdot)$ is, from (33) with $n = 1$,

$$\begin{aligned} &J_{\boldsymbol{\alpha}}(\mathbf{0},\beta_1;\mathbf{y},x_1) \\ &\quad = J(\mathbf{0},\beta_1;\mathbf{y},x_1) \\ &\qquad \times \prod_{i=1}^{m}\left(\lambda(y_i)+\int_S h(s)p(y_i \mid s)P^D(s)f(s)ds\right. \\ &\qquad\qquad \left. +\beta_1 p(y_i \mid x_1)P^D(x_1)f(x_1)\right). \end{aligned} \tag{36}$$

The derivative of the logarithm of (36) with respect to β_1 evaluated at $\beta_1 = 0$ is

$$\begin{aligned} &m_{[1]}(x_1) \\ &\quad = \frac{1}{J_{\boldsymbol{\alpha}}(\mathbf{0},0;\mathbf{y},x_1)}\frac{d}{d\beta_1}J_{\boldsymbol{\alpha}}(\mathbf{0},0;\mathbf{y},x_1) \\ &\quad = (1-P^D(x_1))f(x_1) + \sum_{i=1}^{m}\frac{p(y_i \mid x_1)P^D(x_1)f(x_1)}{\lambda(y_i)+\int_S p(y_i \mid s)P^D(s)f(s)ds}. \end{aligned} \tag{37}$$

The expression (37) is the PHD filter information update.

5.2. Example 2. Pair-Correlation Function of the Bayes Posterior Target Process

"Spooky action at a distance" [4] is a source of concern using point process models for tracking independent targets. One cause is nontrivial pair-correlation [1] in the Bayes posterior process. It is shown in this example that the Bayes posterior target process is repulsive for all x_1 and x_2. This result was first derived in [2].

For Example 1 the second factorial moment is given by the normalized second derivative (31) of the secular function. The derivative $J_\alpha(\cdot)$ is, using (33) with $n = 2$,

$$\begin{aligned} &J_{\boldsymbol{\alpha}}(\mathbf{0},\boldsymbol{\beta};\mathbf{y},\mathbf{x}) \\ &\quad = J(\mathbf{0},\boldsymbol{\beta};\mathbf{y},\mathbf{x})\prod_{i=1}^{m}\left(\lambda(y_i)+\int_S h(s)p(y_i \mid s)P^D(s)f(s)ds\right. \\ &\qquad\qquad \left. +\sum_{j=1}^{2}\beta_j p(y_i \mid x_j)P^D(x_j)f(x_j)\right). \end{aligned} \tag{38}$$

The first derivative of the logarithm of (38) with respect to β_ℓ, $\ell = 1,2$, is

$$\begin{aligned} &\frac{1}{J_{\boldsymbol{\alpha}}(\mathbf{0},\boldsymbol{\beta};\mathbf{y},\mathbf{x})}\frac{d}{d\beta_\ell}J_{\boldsymbol{\alpha}}(\mathbf{0},\boldsymbol{\beta};\mathbf{y},\mathbf{x}) \\ &\quad = (1-P^D(x_\ell))f(x_\ell) + \sum_{i=1}^{m}\frac{p(y_i \mid x_\ell)P^D(x_\ell)f(x_\ell)}{\lambda(y_i)+\int_S p(y_i \mid s)P^D(s)f(s)ds+\sum_{j=1}^{2}\beta_j p(y_i \mid x_j)P^D(x_j)f(x_j)}. \end{aligned} \tag{39}$$

Now let $\ell = 1$ in (39) and take the first derivative with respect to β_2. Setting $\beta_1 = \beta_2 = 0$, rearranging terms, and substituting the first moment (37) evaluated at the points x_1 and x_2 of $\mathbf{x}$ gives

$$\frac{1}{J_{\boldsymbol{\alpha}}(\mathbf{0},\mathbf{0};\mathbf{y},\mathbf{x})}\frac{d^2}{d\beta_1 d\beta_2}J_{\boldsymbol{\alpha}}(\mathbf{0},\mathbf{0};\mathbf{y},\mathbf{x}) = m_{[1]}(x_1)m_{[1]}(x_2) - \sum_{i=1}^{m}\frac{p(y_i \mid x_1)P^D(x_1)f(x_1)p(y_i \mid x_2)P^D(x_2)f(x_2)}{\left(\lambda(y_i) + \int_S p(y_i \mid s)P^D(s)f(s)ds\right)^2}. \tag{40}$$

From (31) the left hand side of (40) is seen to be the second factorial moment of the PGFL (35) of the Bayes posterior point process. Dividing by the product of first factorial moments gives

$$\rho(x_1,x_2) = 1 - \frac{1}{m_{[1]}(x_1)m_{[1]}(x_2)} \times \sum_{i=1}^{m}\frac{p(y_i \mid s_1)P^D(s_1)f(s_1)p(y_i \mid s_2)P^D(s_2)f(s_2)}{\left(\lambda(y_i) + \int_S p(y_i \mid s)P^D(s)f(s)ds\right)^2}. \tag{41}$$

It is evident that $\rho(x_1,x_2) < 1$; therefore, the Bayes posterior process is repulsive.

5.3. Example 3. Multisensor Multitarget Tracking Filters

Let $L \geq 1$ sensors produce conditionally independent measurements in the spaces Y^ℓ, $\ell = 1,\ldots,L$. Example 1 is the special case $L = 1$. The joint PGFL [9] can be written explicitly as

$$\Psi[g^1,\ldots,g^L,h] = \exp\left(-\sum_{\ell=1}^{L}\int_{Y^\ell}\lambda^\ell(y)dy + \sum_{\ell=1}^{L}\int_{Y^\ell}g^\ell(y)\lambda^\ell(y)dy - \int_S f(s)ds + \int_S h(s)f(s)\prod_{\ell=1}^{L}\left(1 - P^{D^\ell}(s) + P^{D^\ell}(s)\int_{Y^\ell}g^\ell(y)p^\ell(y \mid s)dy\right)ds\right), \tag{42}$$

where the sensor likelihood functions and detection probabilities are given by $p^\ell(y \mid s)$ and $P^{D^\ell}(s)$, respectively.

Let $\boldsymbol{\alpha}^\ell = (\alpha_i^\ell : i = 1,\ldots,m^\ell)$, $\ell = 1,\ldots,L$. The secular function of (42) with respect to impulses at the sensor measurements $\mathbf{y}^\ell = \{y_i^\ell : i = 1,\ldots,m^\ell\}$, $y_i^\ell \in Y^\ell$, and at target state $\mathbf{x} = x_1$ is found by substituting

$$g^\ell(y) = \sum_{i=1}^{m^\ell}\alpha_i^\ell\delta_{y_i^\ell}(y), \quad y \in Y^\ell, \quad \ell = 1,\ldots,L,$$
$$h(s) = 1 + \beta_1\delta_{x_1}(s). \tag{43}$$

We do not discuss the pair-correlation function here, so the perturbation of h requires only one term. The logarithm of the secular function is

$$\log J(\boldsymbol{\alpha}^1,\ldots,\boldsymbol{\alpha}^L,\beta_1;\mathbf{y},x_1) = -\sum_{\ell=1}^{L}\int_{Y^\ell}\lambda^\ell(y)dy + \sum_{\ell=1}^{L}\sum_{i=1}^{m^\ell}\alpha_i^\ell\lambda^\ell(y_i^\ell) - \int_S f(s)ds + \int_S f(s)\prod_{\ell=1}^{L}\left(1 - P^{D^\ell}(s) + P^{D^\ell}(s)\sum_{i=1}^{m^\ell}\alpha_i^\ell p^\ell(y_i^\ell \mid s)\right)ds + \beta_1 f(x_1)\prod_{\ell=1}^{L}\left(1 - P^{D^\ell}(x_1) + P^{D^\ell}(x_1)\sum_{i=1}^{m^\ell}\alpha_i^\ell p^\ell(y_i^\ell \mid x_1)\right), \tag{44}$$

where $\mathbf{y} = (\mathbf{y}^1,\ldots,\mathbf{y}^L)$. For $n = 1$ and $L = 1$, this expression is identical to (33).

Differentiating the secular function (44) with respect to $\boldsymbol{\alpha}^1,\ldots,\boldsymbol{\alpha}^L$ and evaluating it for $\alpha_i^\ell = 0$ gives the symbolic expression for the functional derivative at the point $x_1 \in S$. From (30), the symbolic derivative of the logarithm of this expression evaluated at $\beta_1 = 0$ gives the exact numerical value of the intensity function at the point x_1. Except for trivial cases, these derivatives are unsuited to manual differentiation because they have a prohibitively large number of terms. Symbolic differentiation packages can evaluate the required derivatives in principle, at least for sufficiently small problems. Nonetheless, regardless of the computer, these methods will struggle for larger problems. (The potential of automatic differentiation [5] to help in this problem remains to be studied.)

5.4. Example 4. Extended-Targets and Multiple Sensors

The joint PGFL for extended-targets and $L \geq 1$ sensors is

$$\Psi[g^1,\ldots,g^L,h] = \exp\left(-\sum_{\ell=1}^{L}\int_{Y^\ell}\lambda^\ell(y)dy + \sum_{\ell=1}^{L}\int_{Y^\ell}g^\ell(y)\lambda^\ell(y)dy - \int_S f(s)ds + \int_S h(s)f(s)\prod_{\ell=1}^{L}\Psi^{D^\ell}[g^\ell \mid s]ds\right), \tag{45}$$

where $\Psi^{D\ell}[g^\ell \mid s]$ is the PGFL of the measurement point process for a target at $s \in S$. Let $G^\ell(z)$ denote the generating function of the number $\tau \geq 1$ of target measurements, conditioned on target detection. Thus $\Pr^\ell\{\tau = 0\} = 0$ and

$$G^\ell(z) = \sum_{\tau=1}^{\tau^\ell_{\max}} \Pr^\ell\{\tau\} z^\tau, \tag{46}$$

where z is the complex variable of the generating function and $\tau^\ell_{\max} \geq 1$ is the maximum number of measurements that a target at s can produce in sensor ℓ. Assuming that targets generate i.i.d. measurements in sensor ℓ with PDF $p^\ell(\cdot \mid s)$, the conditional measurement PGFL is

$$\Psi^{D\ell}[g^\ell \mid s] = 1 - P^{D\ell}(s) + P^{D\ell}(s) G^\ell\left(\int_{Y^\ell} g^\ell(y) p^\ell(y \mid s) dy\right). \tag{47}$$

For each $s \in S$, $\Psi^{D\ell}[1 \mid s] = 1$ since, by definition, $G^\ell(1) = 1$.

The special case of (45) for $L = 1$ and Poisson generating functions is discussed in [8] and the references therein. The PGFL for $L = 1$ using a more general target point process is given in [11].

Since targets can generate at most $\tau^\ell_{\max}$ measurements, $G^\ell(z)$ is a polynomial of degree $\tau^\ell_{\max}$. If $\tau^\ell_{\max} = 1$, targets generate at most one measurement with probability $P^{D^\ell}(s)$, so $G^\ell(z) \equiv z$ and (45) reduces to the PGFL (44) of Example 3. Another common model is that the number of measurements is Poisson distributed with mean μ_ℓ, conditioned on at least one measurement, so the generating function is $G^\ell(z) = (e^{\mu_\ell z} - 1)/(e^{\mu_\ell} - 1)$.

The joint PGFL for multiple sensors and multiple target measurements is determined by substituting (47) into (45). If only the intensity function is evaluated, the perturbation of h is limited to one term, and the secular function can be found using the same perturbations as (43). The secular function of $\Psi[g^1, \ldots, g^L, h]$ is then

$$\begin{aligned}\log J(\boldsymbol{\alpha}^1, \ldots, \boldsymbol{\alpha}^L, \beta_1; \mathbf{y}, x_1) = &- \sum_{\ell=1}^{L} \int_{Y^\ell} \lambda^\ell(y) dy + \sum_{\ell=1}^{L} \sum_{i=1}^{m^\ell} \alpha^\ell_i \lambda^\ell(y^\ell_i) - \int_S f(s) ds \\ &+ \int_S f(s) \prod_{\ell=1}^{L} \left(1 - P^{D\ell}(s) + P^{D\ell}(s) G^\ell\left(\sum_{i=1}^{m^\ell} \alpha^\ell_i p^\ell(y^\ell_i \mid s)\right)\right) ds \\ &+ \beta_1 f(x_1) \prod_{\ell=1}^{L} \left(1 - P^{D\ell}(x_1) + P^{D\ell}(x_1) G^\ell\left(\sum_{i=1}^{m^\ell} \alpha^\ell_i p^\ell(y^\ell_i \mid x_1)\right)\right).\end{aligned} \tag{48}$$

Derivatives of the secular function can be found by differentiating as needed under the integral sign (absolute convergence holds). As in Example 3, the numerical integrals can be calculated by summing the integrands over the current particle set.

6. DERIVATIVES OF SECULAR FUNCTIONS

The natural way to use secular functions in most applications is to find the exact symbolic derivatives using a software package for ordinary differentiation. Such software is often organized so that the symbolic derivative can be evaluated numerically at specified points, e.g., the particles in a particle filter, by exploiting the internal software representation of the derivative. This calculation bypasses the need to recode (or even to examine) the symbolic expressions.

Automatic differentiation (AD) methods are relatively new [5] techniques in which the numerical values of the symbolic derivative of a function are found without finding the symbolic derivative. These are exact methods (to machine precision), not approximations. Moreover, the additional computational effort is proportional to that of evaluating the function alone. AD methods are based on the chain rule. Their potential use for tracking applications is outside the scope of the present paper.

Alternatively, it may be worthwhile in some applications to consider classical numerical approximations of the symbolic derivatives. Two such methods, finite differences and Maclaurin series expansion, are briefly considered in this section for computing the intensity function.

6.1. Method 1: Classical Finite Differences

In the examples of Section V the derivatives of the secular function with respect to β_1 evaluated at $\beta_1 = 0$ can be evaluated easily for any $\boldsymbol{\alpha} = (\boldsymbol{\alpha}^1, \ldots, \boldsymbol{\alpha}^L) \in \mathbb{R}^M$, where $M = m^1 + \cdots + m^L$ is the total number of sensor measurements. Complexity grows with the number of derivatives with respect to $\boldsymbol{\alpha}$. The intensity function of the Bayes posterior point process is

$$m_{[1]}(x) = \frac{\left(\dfrac{d^M}{d\alpha_1 \cdots d\alpha_M} J'(\alpha, 0)\right)_{\alpha_1 = \cdots = \alpha_M = 0}}{\left(\dfrac{d^M}{d\alpha_1 \cdots d\alpha_M} J(\alpha, 0)\right)_{\alpha_1 = \cdots = \alpha_M = 0}}. \tag{49}$$

The derivatives in (49) can be approximated by classical finite differences.

For real valued functions $U : \mathbb{R}^M \to \mathbb{R}$, the (symmetric) finite difference approximation to the cross-derivative of U at the origin $(0, \ldots, 0) \in \mathbb{R}^M$ is

$$\frac{d^m U}{dx_1 \cdots dx_M}(0,\ldots,0)$$
$$\cong \frac{1}{\varepsilon_1 \cdots \varepsilon_M 2^M} \sum_{\sigma_1,\ldots,\sigma_M=0}^{1} (-1)^{\sigma_1+\cdots\sigma_M} U((-1)^{\sigma_1}\varepsilon_1,\ldots, (-1)^{\sigma_M}\varepsilon_M), \quad (50)$$

where the increments ε_j are suitably "small." In words, the sum is over all 2^M combinations of signs $\{+1,-1\}$. The constant $\varepsilon_1 \cdots \varepsilon_M 2^M$ in (50) cancels out of the ratio (49), leaving only the sums in numerator and denominator. The alternating signs in the sum (50) can lead to underflow for sufficiently small increments. Underflow can be reduced by accumulating the positive terms and negative terms separately and then taking the difference.

The number of terms in the finite difference form (50) is not impractical for small values of M. For values up to, say, $M = 10$, the difficulties encountered can be mitigated by fast multi-core computers. Whatever the limiting value of M, it is ultimately necessary to restrict the measurement space to one or more "windows" that contain at most M measurements.

6.2. Method 2: Maclaurin Series Expansion

PGFLs and their secular functions encode combinatorial information. Consequently, truncating any series approximation to them can be equivalent to a combinatorial constraint. Truncating the Maclaurin series after the linear term is shown to be such a case this subsection.

The secular functions in the examples of Section V have the form, for some choice of constant s and vector $\mathbf{c} \in \mathbb{R}^M$,

$$J(\boldsymbol{\alpha}, \beta_1; \mathbf{y}, x_1) = c_0 \exp(\mathbf{c}^T \boldsymbol{\alpha} + \pi(\boldsymbol{\alpha}) + \beta_1 \pi(\boldsymbol{\alpha}; x_1)), \quad (51)$$

where the function $\pi : \mathbb{R}^M \times S \to \mathbb{R}$ and $\pi(\boldsymbol{\alpha}) \equiv \int_S \pi(\boldsymbol{\alpha}; s)\, ds$. Expanding the integrand $\pi(\boldsymbol{\alpha}; x)$ in a Maclaurin series gives

$$\pi(\boldsymbol{\alpha}; s) \cong \pi(\mathbf{0}; s) + [\nabla \pi(\mathbf{0}; s)]^T \boldsymbol{\alpha} + \tfrac{1}{2} \boldsymbol{\alpha}^T [\nabla^2 \pi(\mathbf{0}; s)] \boldsymbol{\alpha} + \cdots, \quad (52)$$

where $\nabla \pi(\mathbf{0}; s) \equiv (\nabla \pi_\ell(\mathbf{0}; s) : \ell = 1, \ldots, M) \in \mathbb{R}^M$ and $\nabla^2 \pi(\mathbf{0}; s) \in \mathbb{R}^M \times \mathbb{R}^M$ are the gradient and Hessian matrix of $\pi(\boldsymbol{\alpha}; s)$, respectively, evaluated at $\boldsymbol{\alpha} = \mathbf{0} \in \mathbb{R}^M$. Substituting (52) into (51) and retaining only linear terms gives the approximate secular function

$$J(\boldsymbol{\alpha}, \beta_1; \mathbf{y}, x_1) \cong c_0 \exp\left(\mathbf{c}^T \boldsymbol{\alpha} + \int_S [\nabla \pi(\mathbf{0}; s)]^T \boldsymbol{\alpha}\, ds + \beta_1 [\pi(\mathbf{0}; x_1) + [\nabla \pi(\mathbf{0}; x_1)]^T \boldsymbol{\alpha}] \right). \quad (53)$$

It follows from (30) that the intensity function is

$$m_{[1]}(x_1) = \pi(\mathbf{0}; x_1) + \sum_{\ell=1}^{M} \frac{\nabla \pi_\ell(\mathbf{0}; x_1)}{\mathbf{c}_\ell + \int_S \nabla \pi_\ell(\mathbf{0}; s) ds}. \quad (54)$$

A close examination of the expression (54) for, say, Example 3, shows that truncating the Maclaurin series to linear terms is tantamount to the combinatorial restriction that a target generate at most one measurement in at most one sensor. This constraint is not realistic in many problems.

Truncating the Maclaurin series after the quadratic or higher order term will result in different combinatorial restrictions, the nature of which is not studied here. The derivatives of these higher order expansions can be evaluated numerically for use in particle filter implementations to evaluate performance.

7. CONCLUSIONS

It is shown that functional derivatives of the PGFL are equivalent to ordinary derivatives of secular functions. Symbolic derivatives of secular functions can be found with widely available software. The secular function technique yields exact, not approximate, values of the functional derivatives of the PGFL. It lends itself to particle filter implementations because particle weights can be found by evaluating derivatives of the secular function, not functional derivatives of the PGFL. Embedding symbolic differentiation software in production code is somewhat unorthodox, but well within modern computing capabilities.

REFERENCES

[1] A. Baddeley
Spatial point processes and their applications.
in *Stochastic Geometry*, Lecture Notes in Mathematics, Vol. 1892, W. Weil, Editor. Berlin: Springer-Verlag, 2007, pp. 1–75.

[2] Ö. Bozdogan, R. Streit, and M. Efe
Reduced Palm Intensity for Track Extraction.
in *Proceedings of the International Conference on Information Fusion*, Istanbul, July 2013.

[3] D. J. Daley and D. Vere-Jones
An Introduction to the Theory of Point Processes, Volume I: Elementary Theory and Methods.
New York: Springer, 1988 (Second Edition, 2003).

[4] D. Fränken, M. Schmidt, and M. Ulmke
"Spooky action at a distance" in the cardinalized probability hypothesis density filter.
IEEE Transactions on Aerospace and Electronic Systems, Vol. 45, no. 4, 2009, 1657–1664.

[5] A. Griewank, L. Lehmann, H. Leovey, and M. Zilberman
Automatic Evaluations of Cross-Derivatives.
Mathematics of Computation, Vol. 83, No. 285, January 2014, 251–274.

[6] M. J. Lighthill
Introduction to Fourier Analysis and Generalized Functions.
Cambridge, UK: Cambridge University Press, 1958.

[7] R. P. S. Mahler
Statistical Multisource-Multitarget Information Fusion.
Norwood MA: Artech House, 2007.

[8] R. P. S. Mahler
PHD Filters for Nonstandard Targets, I: Extended Targets.
in *Proceedings of the 12th International Conference on Information Fusion*, Seattle, Washington, USA, July 2009.

[9] R. P. S. Mahler
Approximate Multisensor CPHD and PHD Filters.
in *Proceedings of the 13th International Conference on Information Fusion*, Edinburgh, Scotland, July 2010.

[10] J. E. Moyal
The General Theory of Stochastic Population Processes.
Acta Mathematica, Vol. 108, 1–31, 1962.

[11] U. Orguner, C. Lundquist, and K. Granström
Extended Target Tracking with a Cardinalized Probability Hypothesis Density Filter.
Department of Electrical Engineering, Linköpings Universitet, Technical report LiTH-ISY-R-2999, 14 March 2011.

[12] R. L. Streit
Poisson Point Processes—Imaging, Tracking, and Sensing.
New York: Springer, 2010.

[13] R. L. Streit
The Probability Generating Functional for Finite Point Processes, and Its Application to the Comparison of PHD and Intensity Filters.
Journal on Advances in Information Fusion, Vol. 8, No. 2, December 2013.

Roy Streit is currently a Senior Scientist at Metron in Reston, Virginia. His research interests include multi-target tracking, multi-sensor data fusion, distributed systems, medical imaging, and signal processing, as well as statistical methods for pharmacovigilance and business analytics. Prior to joining Metron in 2005, he was in the Senior Executive Service at the Naval Undersea Warfare Center in Newport, RI, working primarily on the development, evaluation and application of multi-sensor data fusion algorithms in support of submarine sonar and combat control automation. He was an Exchange Scientist at the Defence Science and Technology Organisation (DSTO) in Adelaide, Australia, from 1987–1989. In 1999, he received the Solberg Award from the American Society of Naval Engineers for contributions to naval engineering through personal research, and in 2001 the Department of the Navy Superior Civilian Achievement Award. From 1996 to 2005 he served on the Sonar Technology Panel (Panel 9) of The Technical Cooperation Program (TTCP), a multinational governmental organization supporting scientific information exchange between member nations. He was President of the International Society for Information Fusion (ISIF) in 2012, and he continues to serve on the ISIF Board of Directors.

Dr. Streit is a Fellow of the IEEE. He is the author of a book entitled *Poisson Point Processes: Imaging, Tracking, and Sensing* published by Springer in 2010. It was translated into Chinese and published by Science Press, Beijing, in 2013. He has published papers in over a dozen refereed technical journals, and given numerous invited and contributed papers at international conferences and workshops. He holds nine U.S. patents. He is a Professor (Adjunct) in the Department of Electrical and Computer Engineering at the University of Massachusetts–Dartmouth. He received a Ph.D. in Mathematics in 1978 from the University of Rhode Island. He was a Visiting Scientist at Yale University from 1982–1984, and a Visiting Scholar at Stanford University from 1981–1982. He is currently an Affiliate Consultant with SUCCEED Educational Consultancy based in Rhode Island.

Dynamic Surface Reconstruction by Recursive Fusion of Depth and Position Measurements

GERHARD KURZ
UWE D. HANEBECK

Surface estimation can be performed based on position or depth measurements. We propose a method to fuse both types of measurements. Position measurements are obtained from landmarks on the surface, i.e., they are fixed to a certain point on the surface. In contrast, depth measurements reflect the depth measured along a line emanating from a depth camera and are not fixed to a position on the surface. The proposed approach uses a mixture of Cartesian and polar or spherical coordinate to treat both measurement types accordingly. By doing so, the uncertainties associated with the different measurement types are explicitly considered. The presented method represents the surface by a spline and is applicable to both 2D and 3D applications. Surface estimation is considered as a recursive filtering problem and standard nonlinear filtering methods such as the unscented Kalman filter can be used to obtain surface estimates. We show a thorough evaluation of the proposed approach in simulations.

This is an extended version of the paper "Recursive Fusion of Noisy Depth and Position Measurements for Surface Reconstruction" [18] published at the *16th International Conference on Information Fusion (Fusion 2013)*, which received the *Jean-Pierre Le Cadre Award for Best Paper*.

Manuscript received July 31, 2013; revised December 12, 2013; released for publication December 19, 2013.

Refereeing of this contribution was handled by Peter Willett.

Authors' addresses: Intelligent Sensor-Actuator-Systems Laboratory (ISAS), Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology (KIT), Germany, E-mail: (gerhard.kurz@kit.edu, uwe.hanebeck@ieee.org).

1. INTRODUCTION

Many applications require the reconstruction of surfaces based on noisy measurements. For example, in various medical applications the surface of organs needs to be reconstructed from measurements that originate from medical imaging technologies [22], [6], [29], [7]. Surface reconstruction is also a relevant topic in other areas such as robotics [23] and computer graphics [22], [11].

A practical application where the presented approach may be used is intra-operative beating heart tracking for robot-assisted coronary artery bypass graft. The idea behind this application has first been introduced by Nakamura et al. in 2001 [24] and can be summarized as follows. The surgery is carried out by a robot that is remotely controlled by a surgeon. During the surgery, the movement of the heart is observed by sensors and this information is used to control the robot in such a way that it automatically compensates for the heart motion. The surgeon is in turn presented with a stabilized image [17] of the heart and experiences the illusion of operating on the still heart, which is a significantly easier task. Because operating on the beating heart is very difficult, currently operations are usually performed on the stopped heart, which incurs significant disadvantages for the patient's health, such as a risk of anemia and cerebral microembolization [16]. In order to make robot-assisted beating heart surgery feasible, an accurate reconstruction of the moving heart surface is required.

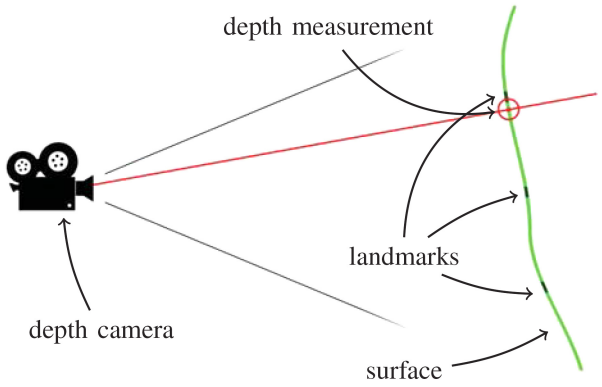


Fig. 1. The considered setting: A depth camera observes a deformable surface (light green) and obtains depth measurements (red circle) along a line emanating from the camera (red line). Additionally, some sparse landmarks (dark green) on the surface can be tracked by other means (e.g., a stereo camera system).

For reconstructing a surface, we consider two different types of measurements (see Fig. 1). First, there are position measurements originating from certain points located at a fixed positions on the surface. Position measurements are typically obtained from landmarks on the surface, for example structured regions that allow 3D reconstruction with a stereo camera system. Second, there are depth measurements that do not originate from a fixed point on the surface and only depend on the dis-

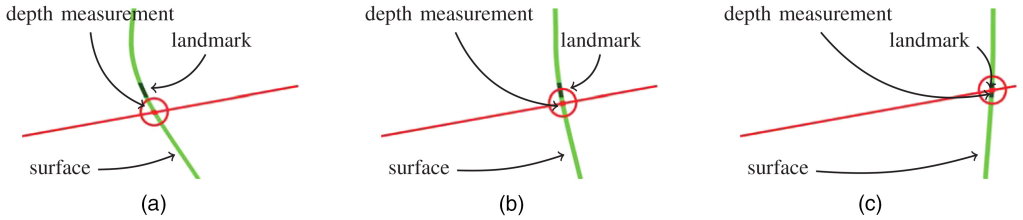


Fig. 2. Depth measurements (red circles) and position measurements of a time-varying surface. Notice how the depth measurement is obtained from different surface points as the surface deforms. (a) Beginning. (b) Middle. (c) End.

tance of a certain point in space to the surface along a given line. Depth measurements can be obtained from depth sensors such as time-of-flight (TOF) cameras or sensors based on structured light such as the Microsoft Kinect. The difference between position and depth measurements is illustrated in Fig. 2.

While position measurements are typically sparse but highly accurate, depth measurements tend to be more plentiful, but less accurate and more susceptible to noise. On the one hand, stereo camera systems may have a high resolution, but perform poorly in non-structured areas. On the other hand, TOF sensors can handle uniform surface areas, but have comparatively limited resolution and accuracy. Thus, it is beneficial to combine both types of measurements in order to achieve a more accurate and robust reconstruction of the surface compared to scenarios where only one type of measurement is used. Fusion of information from different types of sensors allows to alleviate the disadvantages of any given sensor type.

Many practical applications are not limited to a static scenario because the sensors and the surface move relatively to each other. Furthermore, the surface may deform and change shape over time. Consequently, our goal is to track surface position and shape over time and to include new information recursively as it is obtained. Prior knowledge may be included to predict the future evolution of the surface.

We now outline our main contribution. In this paper, we introduce a novel method for surface reconstruction suitable for both 2D and 3D applications. The proposed method combines depth and position measurements to recursively estimate the state of the surface while considering measurement uncertainties. It does not depend on a particular choice of sensor and can be employed in a wide area of applications. Our method is based on a spline representation of the surface whose parameters are recursively estimated using nonlinear filtering techniques. Separate measurement equations for depth and position measurements are derived in order to deal with their individual characteristics.

1.1. Structure

The paper is structured as follows. In Sec. 2, we give an overview of previous work in the area of surface reconstruction. The required prerequisites are introduced in Sec. 3. The presented method is derived for the 2D case in Sec. 4 and adapted to the 3D case in Sec. 5. We propose some further enhancements in Sec. 6 and evaluate the proposed algorithms in simulations in Sec. 7. Finally, we form a conclusion in Sec. 8.

Compared to our previous publication [18], we have significantly extended Sec. 2 to give a more complete overview of literature on the topic and Sec. 6, where we give enhancements to the proposed method that allow the incorporation of angular uncertainties and the use of approximation rather than interpolation functions. We have also performed an evaluation of the adaptive addition of nodes, which is given in Sec. 7. Furthermore, we now provide additional explanations in various places as well as a number of supplementary figures to illustrate the proposed method.

1.2. Notation

We denote vectors by underlined letters $\underline{x}$, matrices by bold letters $\mathbf{A}$, and angles by Greek letters α.

n	number of dimensions ($n = 2$ or 3)
k	time index
$\underline{p}_1, \ldots, \underline{p}_m$	points in $\mathbb{R}^n$
$f_1, \ldots, f_m$	function values in $\mathbb{R}$
$f(\cdot)$	interpolating function
$\phi(\cdot)$	radial basis function
$c_1, \ldots, c_m$	coefficients in interpolating function
interpolate$(\underline{p}_k, f_k)$	interpolation algorithm that returns interpolating function
$\underline{x}_k$	system state in $\mathbb{R}^q$
$\underline{x}_k^e, \underline{x}_k^p$	estimated and predicted state
$a_k(\cdot)$	system function
u_k	system noise with covariance $\mathbf{C}_k^u$
l	number of landmarks
$x_k^{a,b}$	position of landmark a in dimension b
$\hat{\underline{y}}_k$	position measurements
$\underline{v}_k$	position measurement noise with covariance $\mathbf{C}_k^v$
$\hat{\underline{z}}_k$	depth measurements
$\underline{w}_k$	depth measurement noise with covariance $\mathbf{C}_k^w$
$\alpha_1, \ldots, \alpha_r$	measurement angles
$s_k(\cdot)$	interpolated surface
$s_k^{\text{true}}(\cdot)$	true surface

$\phi_1, \ldots, \phi_d$	angles for additional control points in 2D
$(\phi_i, \theta_i)_{i=1,\ldots,d}$	angles for additional control points in 3D
$x_k^{1,*}, \ldots, x_k^{d,*}$	depth of additional control points
σ_{add}	initial standard deviation for additional control points
$E_k^\tau(\cdot)$	RMSE at given angle over τ time steps ending at k
$\delta_k^1, \ldots, \delta_k^r$	noise in angular domain with covariance $\mathbf{C}_k^\delta$ in 2D
$(\delta_k^i, \eta_k^i)_{i=1,\ldots,r}$	noise in angular domain with covariance $\mathbf{C}_k^{\delta,\eta}$ in 3D
λ	relaxation value for approximation
$\gamma_1, \ldots, \gamma_e$	evaluation angles
E_k	RMSE at evaluation angles

2. RELATED WORK

Traditional methods for surface reconstruction exclusively rely on position measurements. For example, Hoppe et al. presented a method to reconstruct a surface based on unorganized points [11]. As a result of the increasingly widespread use of depth cameras, algorithms exclusively based on depth measurements have been introduced, such as Kinect Fusion [13], [25].

However, approaches that try to combine both types of measurements are still fairly new. An early approach was published by Lindner et al. in 2007 [20]. This approach combines information from a TOF camera and a binocular camera to obtain a high-resolution colored point cloud. However, the color information is not used to obtain a more accurate estimate of the 3D shape of the observed scene.

Gudmundsson et al. [9] proposed a fusion algorithm for disparity maps obtained from stereo reconstruction and depth information obtained by TOF cameras in order to obtain higher quality disparity maps. The fusion is achieved by converting the TOF depth values to disparities in the image frames. A more sophisticated approach to the same problem has been presented by Zhu et al. in 2011 [32]. It is based on Markov Random Fields that describe the depth information and can be used to probabilistically combine measurements from a stereo camera system and a time-of-flight (TOF) camera. In the fusion process, the respective uncertainties of both sensor types are considered and their influence is weighted accordingly.

In 2008, Guan et al. [8] proposed an algorithm to combine images from several conventional cameras and a TOF camera for the purpose of 3D object reconstruction. The conventional cameras are used to provide silhouette information and allow construction of the visual hull, whereas the TOF camera is able to obtain depth information in areas where concavities occur. A probabilistic space occupancy grid, i.e., a voxel-based approach was used to obtain the object shape by calculating iso-probability surfaces with a graph-cut algorithm.

TABLE I
Comparison of fusion approaches.

Method	representation	stochastic	recursive
Lindner [20]	RGBD data	no	no
Gudmundsson [9]	disparity map	no	no
Zhu [32]	disparity map	yes	no
Guan [8]	voxel	yes	no
Groch [7]	voxel	yes	no
proposed	spline	yes	yes

Groch et al. [7] have applied a very similar approach to a medical application, but they fused stereo disparities instead of silhouette information.

There are different ways of representing the reconstructed surface. Depth sensors usually provide depth maps or point clouds as raw data, but a more sophisticated representation is desired. One common approach is to use spatial discretization and represent the surface as voxels [13], [8]. However, voxel-based approaches typically require a lot of memory and computational power if a high resolution is to be achieved. Another common approach is to use triangular meshes [22]. While triangular meshes are usually more computationally efficient than voxels, they require a large number of triangles to provide a satisfactory reconstruction of rounded shapes. An alternative is to describe the surface as a spline, which can be stored in a very compact form as it is uniquely defined by a small number of control points [19], [2], [1]. Splines are very suitable for smooth surfaces without rough edges. Unlike voxels or triangular meshes, splines can be evaluated at an arbitrary resolution and still appear smooth. Stochastic formulations of splines have previously been used by Brunn et al. [5]. Gaussian processes can also be used to model uncertain surfaces [26]. In fact, the spline based interpolation used in the proposed approach could easily be replaced with a Gaussian process regression.

Furthermore, we have to distinguish between recursive approaches that are able to include new information as it is obtained and methods that only use information from a single time step. Kinect Fusion [13] considers point clouds at consecutive time steps and tries to align them with the well-known iterative closest point algorithm [4]. The proposed algorithm is also capable of gaining more information over time because it is based on nonlinear recursive filtering methods. However, most approaches in this area are not recursive, i.e., they only consider a single time step.

An overview of the mentioned approaches is given in Table I. It should be noted that some approaches are based on stochastic foundations, e.g., Markov Random Fields, whereas others do not explicitly consider stochastic uncertainties.

3. PREREQUISITES

Before describing the proposed method for surface reconstruction, we introduce some prerequisites.

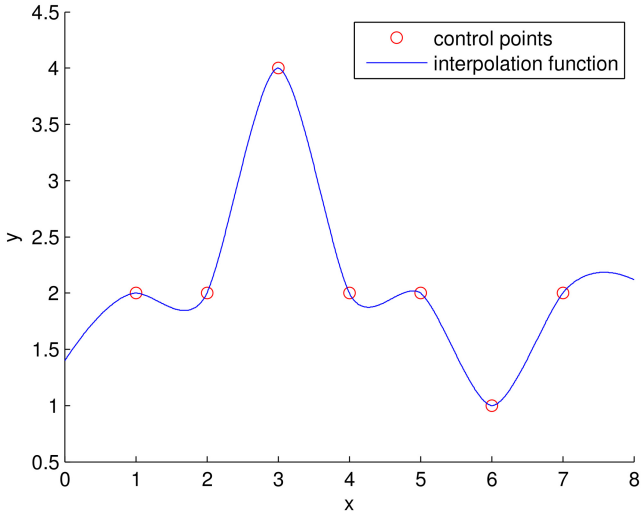


Fig. 3. Example of the interpolation achieved by the presented algorithm for $\mathbb{R} \to \mathbb{R}$ interpolation, i.e., for $n = 1$. The control points are $p_1,\ldots,p_7 = 1,2,3,4,5,6,7$ with values $f_1,\ldots,f_7 = 2,2,4,2,2,1,2$ and the basis function $\phi(x)$ is a thin plate spline.

3.1. Interpolation

Let $m \in \mathbb{N}_{>0}$, $\underline{p}_1,\ldots,\underline{p}_m \in \mathbb{R}^n$ and $f_1,\ldots,f_m \in \mathbb{R}$. The goal of interpolation is to find a function $f : \mathbb{R}^n \to \mathbb{R}$ with $f(\underline{p}_i) = f_i$ for $1 \le i \le m$ where f is smooth in some sense. There are various types of interpolation functions. An overview can be found in [1]. The proposed method does not depend on a particular choice of interpolation method. For later use, we define a function

$$\text{interpolate: } ((\mathbb{R}^n)^m \times \mathbb{R}^m) \to (\mathbb{R}^n \to \mathbb{R}),$$

$$(\underline{p}_1,\ldots,\underline{p}_m; f_1,\ldots,f_m) \mapsto f$$

that maps points $\underline{p}_1,\ldots,\underline{p}_m$ and values $f_1,\ldots,f_m$ to their interpolating function $f \in (\mathbb{R}^n \to \mathbb{R})$.

For the purpose of our experiments, we decided to use Radial Basis Functions (RBF) [6], [2] for interpolation, because they are easy to calculate by solving a system of linear equations and are applicable for any dimension n. The interpolating function f is given by

$$f(\underline{p}) = \sum_{j=1}^{m} c_j \cdot \phi(\|\underline{p} - \underline{p}_j\|),$$

where $\phi : \mathbb{R}_{\ge 0} \to \mathbb{R}$ is the basis function, $c_1,\ldots,c_m \in \mathbb{R}$ are weighting coefficients, and $\|\cdot\|$ is the Euclidean norm. A popular choice of basis function is the thin plate spline (TPS):

$$\phi(x) = \begin{cases} x^2 \log x, & x > 0 \\ 0, & x = 0 \end{cases}.$$

Because the value of $\log(0)$ is undefined, we set $\phi(0) = \lim_{x\to 0}(x^2 \log(x)) = 0$. The reason why the TPS is commonly used as a basis function, is the fact that the interpolation function is C^1 continuous and that it minimizes the energy functional

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \left(\frac{\partial^2 f}{\partial x^2}\right)^2 + 2\left(\frac{\partial^2 f}{\partial x \partial y}\right)^2 + \left(\frac{\partial^2 f}{\partial y^2}\right)^2 dx\,dy,$$

which means that it provides in a certain sense the smoothest possible interpolation function.

The weighting coefficients $c_1,\ldots,c_m$ can be obtained by solving the system of m linear equations

$$f_i = \sum_{j=1}^{m} c_j \cdot \phi(\|\underline{p}_i - \underline{p}_j\|), \quad 1 \le i \le m.$$

The algorithm is given in Algorithm 1. We show an example of the interpolation produced by this algorithm in Fig. 3.

ALGORITHM 1 *Interpolation based on RBFs.*

Input: radial basis function $\phi : \mathbb{R}_{\ge 0} \to \mathbb{R}$;
points $p_1,\ldots,p_m \in \mathbb{R}^n$;
values $f_1,\ldots,f_m \in \mathbb{R}$

Output: interpolation function $f : \mathbb{R}^n \to \mathbb{R}$

// prepare matrix of RBF values
$\mathbf{A} \leftarrow (m \times m$ matrix);
for $i \leftarrow 1$ **to** m **do**
  for $j \leftarrow i$ **to** m **do**
    $\mathbf{A}(i,j) \leftarrow \phi(\|p_i - p_j\|)$;
    // $\mathbf{A}$ is symmetric
    $\mathbf{A}(j,i) \leftarrow \mathbf{A}(i,j)$;
  end
end
// solve $\mathbf{A} \cdot [c_1,\ldots c_m]^T = [f_1,\ldots,f_m]^T$
$[c_1,\ldots c_m]^T \leftarrow \mathbf{A}^{-1}[f_1,\ldots,f_m]^T$;
// obtain interpolation function
$f \leftarrow (p \mapsto \sum_{j=1}^{m} c_j \cdot \phi(\|p - p_j\|))$;
return f;

3.2. Polar and Spherical Coordinates

While many common approaches rely on Cartesian coordinates exclusively, we use polar coordinates (in 2D) and spherical coordinates (in 3D) to simplify certain computations similar to [3]. The transformation between Cartesian and polar coordinates is given by

$$x = r\cos(\phi),$$

$$y = r\sin(\phi),$$

and

$$r = \|(x,y)^T\| = \sqrt{x^2 + y^2},$$

$$\phi = \text{atan2}(y,x).$$

For spherical coordinates, there are several common definitions. We use the convention

$$x = r\cos(\theta)\cos(\phi),$$

$$y = r\cos(\theta)\sin(\phi),$$

$$z = r\sin(\theta),$$

and

$$r = \|(x,y,z)^T\| = \sqrt{x^2 + y^2 + z^2},$$

$$\phi = \text{atan2}(y,x),$$

$$\theta = \arcsin(z/r).$$

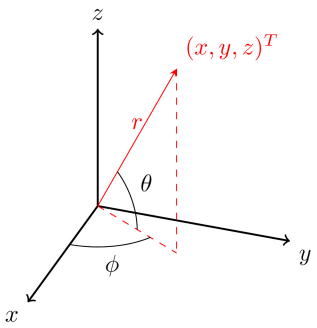


Fig. 4. Spherical coordinates as used in this paper.

An illustration of the meaning of ϕ and θ is given in Fig. 4. Be aware that we use θ as the angle between the x-y-plane and the vector $(x,y,z)^T$ whereas some common definitions use θ as the angle between the z-axis and the $(x,y,z)^T$ vector. It also deserves mentioning that some authors reverse the roles of ϕ and θ.

3.3. System and State Representation

For describing the estimate of the reconstructed surface at time step k, we use the state vector $\underline{x}_k^e \in \mathbb{R}^q$. We assume the state to be Gaussian-distributed with covariance matrix $\mathbf{C}_k^e$. A system model

$$\underline{x}_{k+1}^p = a_k(\underline{x}_k^e) + \underline{u}_k$$

with system function $a_k : \mathbb{R}^q \to \mathbb{R}^q$ and additive zero-mean Gaussian noise $\underline{u}_k \sim \mathcal{N}(\underline{0}, \mathbf{C}_k^u)$ can be used to describe the evolution of the state $\underline{x}_k$.

The structure of the system with the estimator is depicted in Fig. 5. If the system model is linear, the Kalman filter formulas [15] can be used to perform the prediction step. Otherwise a nonlinear filter such as the unscented Kalman filter (UKF, [14]) can be applied. Note that even if the system model is linear, we still need a nonlinear filter for the depth measurement update.

If the system does not follow any known dynamics, a random walk model may be used. In static cases, where the surface does not change over time, prediction can be omitted. As our approach for estimating the reconstructed surface is independent of the particular details of the system model, we will focus on the measurement model from now on.

4. SURFACE RECONSTRUCTION IN 2D

Let us first consider the 2D case. Although the 2D case might not seem relevant at first, there are actually a number of applications for 2D surface reconstruction. For example, LIDAR (light detection and ranging) sensors are commonly used in robotics and allow the reconstruction of obstacles as surfaces in 2D [30].

4.1. Position Measurements

We consider a set of $l \in \mathbb{N}$ landmarks on the surface. For tracking these landmarks, we define the state vector at time step k as

$$\underline{x}_k = [x_k^{1,1}, x_k^{1,2} \ldots, x_k^{l,1}, x_k^{l,2}]^T \in \mathbb{R}^{2l},$$

where $x_k^{a,b}$ represents the position of landmark $a \in \{1,\ldots,l\}$ in dimension $b \in \{1,2\}$ at time step k. In this case, the measurement model is trivially given by

$$\hat{\underline{y}}_k = \mathbf{I}_{2l\times 2l} \cdot \underline{x}_k + \underline{v}_k,$$

where $\mathbf{I}_{2l\times 2l} \in \mathbb{R}^{2l\times 2l}$ is the identity matrix, $\hat{\underline{y}}_k$ is the measurement at time step k, and $\underline{v}_k$ is additive Gaussian noise with $\underline{v}_k \sim \mathcal{N}(\underline{0}, \mathbf{C}_k^v)$. As the measurement equation

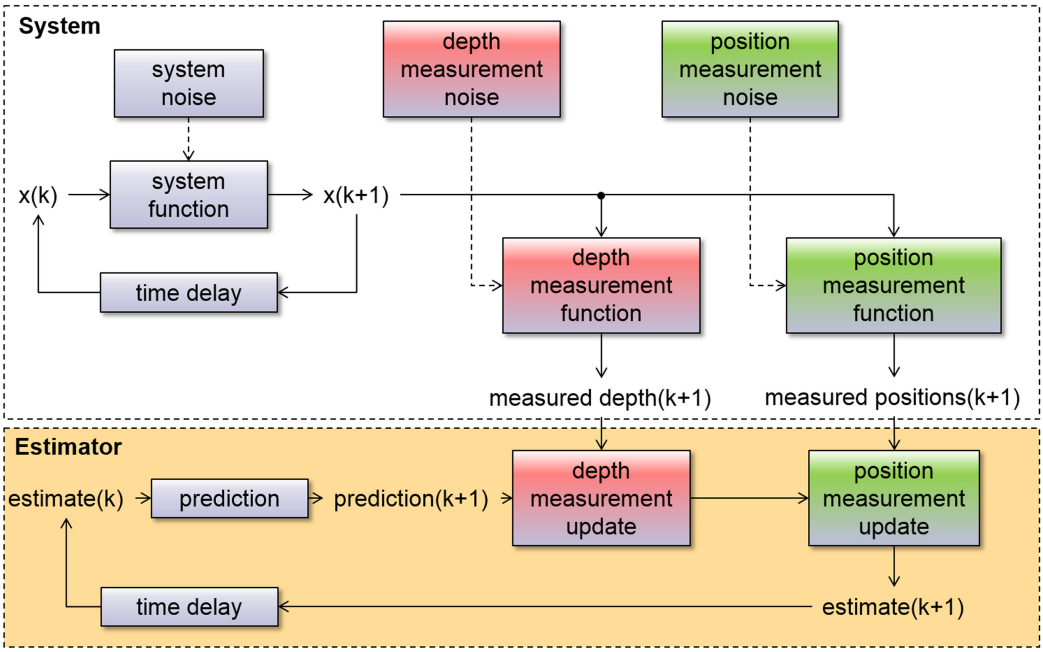


Fig. 5. The structure of the system and the recursive estimator. We propose two separate measurement updates for depth and position measurements.

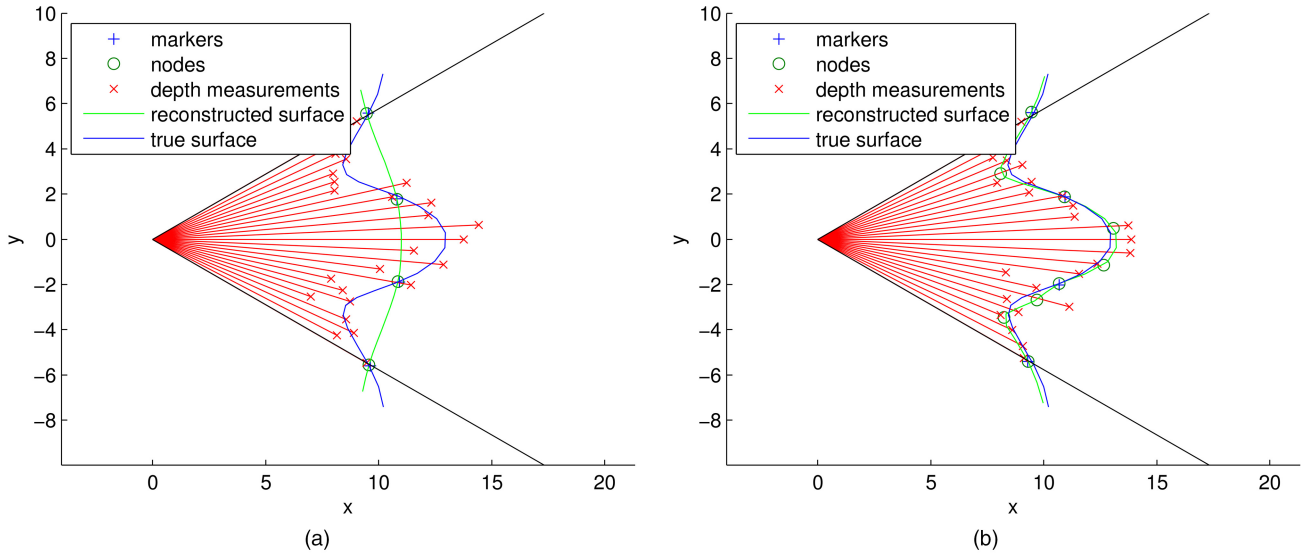


Fig. 6. Reconstructed and true surfaces in 2D. (a) Before state augmentation. (b) After state augmentation.

is linear, a Kalman filter [15] can be used to perform the measurement update. The surface s_k at time step k can be found by performing an interpolation through the currently estimated positions of the landmarks with any suitable interpolation method.

4.2. Depth Measurements

In addition to position measurements, we now want to include depth measurements into the estimation procedure. For the moment, we assume a single depth camera. Without loss of generality, we define it to be located at the origin of the coordinate system and facing towards $[1,0]^T$. We further assume that the depth camera can obtain $r \in \mathbb{N}$ depth measurements $\hat{z}_k^1,\dots,\hat{z}_k^r$ at r different angles $\alpha_1,\dots,\alpha_r$. These angles are typically evenly spread across the depth camera's field of view. Consequently, the measurement equation has to calculate the intersections of the lines at angles $\alpha_1,\dots,\alpha_r$ with the surface. Depending on the surface representation, calculating this intersection can be very difficult. One of the key ideas of our approach is to use polar coordinates, which nicely circumvents this problem. If we parameterize the surface as a function $s_k : \mathbb{R} \to \mathbb{R}$ which maps angles α to distances $s_k(\alpha)$, the intersection for the lines at angles $\alpha_1,\dots,\alpha_r$ are trivially calculated as $s_k(\alpha_1),\dots,s_k(\alpha_r)$.

This yields the measurement equation

$$\underline{\hat{z}}_k = \begin{bmatrix} \hat{z}_k^1 \\ \vdots \\ \hat{z}_k^r \end{bmatrix} = \begin{bmatrix} s_k(\alpha_1) \\ \vdots \\ s_k(\alpha_r) \end{bmatrix} + \underline{w}_k,$$

$$s_k(\alpha) = \text{interpolate } (p_k; f_k)(\alpha),$$

$$p_k = (\text{atan2}(x_k^{1,2}, x_k^{1,1}),\dots,\text{atan2}(x_k^{l,2}, x_k^{l,1})),$$

$$f_k = \left(\left\| \begin{bmatrix} x_k^{1,1} \\ x_k^{1,2} \end{bmatrix} \right\|, \dots, \left\| \begin{bmatrix} x_k^{l,1} \\ x_k^{l,2} \end{bmatrix} \right\| \right)$$

with measurements $\underline{\hat{z}}^k$ and Gaussian noise $\underline{w}_k \sim \mathcal{N}(\underline{0}, \mathbf{C}_k^w)$. Intuitively, the landmark coordinates $(x_k^{i,1}, x_k^{i,2})_{1 \le i \le l}$ are converted to polar coordinates with angular coordinates p_k and radial coordinates f_k. From these polar coordinates, the interpolation function $s_k(\cdot)$ is obtained. By evaluating the interpolation function $s_k(\cdot)$ at the measurement angles $\alpha_1,\dots,\alpha_r$, we calculate the measurement, which is finally disturbed by noise $\underline{w}_k$.

Be aware that w_k not only contains the stochastic error from the noise of the depth camera but also a modeling error as a result of the spline interpolation. The equations for p_k and f_k follow from the conversion of Cartesian into polar coordinates as described in Sec. 3.2. The surface function s_k is derived from $\underline{x}_k$ by interpolation, which is in general nonlinear in $\underline{x}_k$. Thus, it is necessary to use a nonlinear filter to perform the depth measurement update. For example, the UKF [14], the S^{2}KF (Smart Sampling Kalman Filter) [28], or the Gaussian filter introduced in [12] may be used.

4.3. State Augmentation

While it is possible to use depth measurements as described previously, the achievable accuracy is still strongly limited by the number of position measurements. The reason for this issue is the fact that the number of degrees of freedom of the reconstructed surface is determined by the number of position measurements. An example of this limitation is depicted in Fig. 6. Before state augmentation, all four markers are estimated correctly, but the estimated surface is very different from the true surface. After augmenting the state, additional nodes give the surface more degrees of freedom and the true surface can be approximated much more closely.

To improve accuracy, we augment the state by additional control points that do not correspond to landmarks. One may be tempted to augment the state by the Cartesian coordinates of points in $\mathbb{R}^2$, which lie some-

where on the surface, and to try to estimate their position. However, as these points do not produce measurements originating from a fixed position on the surface, their location cannot be uniquely determined from the measurements (see Fig. 2). Any position in space that leads to the same interpolated surface s_k is just as reasonable an estimate as any other. Consequently, the problem is underdetermined and the state is not observable.

The key idea is to introduce additional nodes not as arbitrary points in $\mathbb{R}^2$ but in polar coordinates as $d \in \mathbb{N}$ depths at certain fixed angles $\varphi_1,\dots,\varphi_d$. This yields an augmented state

$$\underline{x}_k = (\underbrace{x_k^{1,1},x_k^{1,2},\dots,x_k^{l,1},x_k^{l,2}}_{\text{landmarks}}, \quad \underbrace{x_k^{1,*},\dots,x_k^{d,*}}_{\text{additional control points}} \quad)^T \in \mathbb{R}^{2l+d},$$

where $x_k^{1,*}\dots x_k^{d,*}$ are the depths at angles $\varphi_1,\dots,\varphi_d$. These angles are not part of the state, as they are not estimated but chosen as fixed values. Consequently the depths $x_k^{1,*}\dots x_k^{d,*}$ are uniquely determined by the shape of the surface and, hence, the state is observable.

This poses the question how the angles $\varphi_1,\dots,\varphi_d$ should be chosen. Simple approaches include random angles inside the camera's field of view or angles that lie on an evenly spaced grid. More sophisticated ways to choose appropriate angles are discussed in Sec. 6.1.

The measurement equation for positions becomes

$$\hat{\underline{y}}_k = [\mathbf{I}_{2l\times 2l}\mathbf{0}_{2l\times d}]\underline{x}_k + \underline{v}_k,$$

with identity matrix $\mathbf{I}_{2l\times 2l} \in \mathbb{R}^{2l\times 2l}$ and zero matrix $\mathbf{0}_{2l\times d} \in \mathbb{R}^{2l\times d}$, which just ignores the additional control points. The measurement equation for depth becomes

$$\hat{\underline{z}}_k = \begin{bmatrix}\hat{z}_k^1\\ \vdots \\ \hat{z}_k^r\end{bmatrix} = \begin{bmatrix}s_k(\alpha_1)\\ \vdots \\ s_k(\alpha_r)\end{bmatrix} + \underline{w}_k,$$

$$s_k(\alpha) = \text{interpolate } (p_k; f_k)(\alpha),$$

$$p_k = (\text{atan2}(x_k^{1,2},x_k^{1,1}),\dots,\text{atan2}(x_k^{l,2},x_k^{l,1}), \varphi_1,\dots,\varphi_d),$$

$$f_k = \left(\left\|\begin{bmatrix}x_k^{1,1}\\ x_k^{1,2}\end{bmatrix}\right\|,\dots,\left\|\begin{bmatrix}x_k^{l,1}\\ x_k^{l,2}\end{bmatrix}\right\|, x_k^{1,*},\dots,x_k^{d,*}\right),$$

which now includes the additional control points in the interpolation process. Once again, the landmark coordinates $(x_k^{i,1},x_k^{i,2})_{1\le i\le l}$ are converted to polar coordinates. Together with the additional control points, whose polar coordinates are $(\phi_i,x_k^{i,*})_{1\le i\le d}$, they form a list of points with angular coordinates p_k and radial coordinates f_k. The surface $s_k(\cdot)$ is calculated by interpolation through all of these points. When augmenting the state, the covariance matrix is augmented as well according to

$$\mathbf{C}_k^e \leftarrow \begin{bmatrix}\mathbf{C}_k^e & \mathbf{0}_{2l\times d}\\ \mathbf{0}_{d\times 2l} & \sigma_{\text{add}}^2\cdot\mathbf{I}_{d\times d}\end{bmatrix}$$

with uncertainty $\sigma_{\text{add}} > 0$.

Augmenting the state vector also involves augmenting the system model. For this purpose, we define an augmented system function $a_k : \mathbb{R}^{2l+d} \to \mathbb{R}^{2l+d}$ that maps the augmented state at time step k to the augmented state at time step $k+1$. The system noise $\underline{u}_k$ is augmented to $2d+l$ dimensions as well. The details of this augmentation depend on the system model that is used for the given application. In case of a random walk model, the additional dimensions can be assumed to conform to a random walk model as well.

5. SURFACE RECONSTRUCTION IN 3D

For many applications that are relevant in practice, 3D surface reconstruction is required. Fortunately, the presented methods can easily be applied to a 3D setting as well.

To accommodate for the third dimension, a few changes are required. Positions in $\mathbb{R}^2$ are replaced with positions in $\mathbb{R}^3$ and polar coordinates are replaced with spherical coordinates. We also change the surface representation to a function $s_k : \mathbb{R}^2 \to \mathbb{R}$ that maps pairs of angles (α,β) to distances $s_k(\alpha,\beta)$. Once again we assume a single depth camera. Without loss of generality, it is located in the origin, its orientation is aligned with the coordinate system, and it is facing towards $[1,0,0]^T$.

These modifications yield the state representation

$$\underline{x}_k = [\underbrace{x_k^{1,1},x_k^{1,2},x_k^{1,3},\dots,x_k^{l,1},x_k^{l,2},x_k^{l,3}}_{\text{landmarks}}, \underbrace{x_k^{1,*},\dots,x_k^{d,*}}_{\text{additional control points}} \quad]^T \in \mathbb{R}^{3l+d},$$

where $x_k^{a,b}$ represents the position of landmark a in dimension $b \in \{1,2,3\}$ at time step k. The angles of the additional control points $(\varphi_1,\theta_1),\dots,(\varphi_d,\theta_d)$ are once again fixed and not part of the state. The measurement equation for positions is now

$$\hat{\underline{y}}_k = [\mathbf{I}_{3l\times 3l}\mathbf{0}_{3l\times d}]\underline{x}_k + \underline{v}_k,$$

and the measurement model for depth is

$$\hat{\underline{z}}_k = \begin{bmatrix}\hat{z}_k^1\\ \vdots \\ \hat{z}_k^r\end{bmatrix} = \begin{bmatrix}s_k(\alpha_1,\beta_1)\\ \vdots \\ s_k(\alpha_r,\beta_r)\end{bmatrix} + \underline{w}_k,$$

$$s_k(\alpha,\beta) = \text{interpolate } (p_k; f_k)(\alpha,\beta),$$

$$p_k = \left(\begin{bmatrix}\text{atan2}(x_k^{1,2},x_k^{1,1})\\ \arcsin\left(\frac{x_k^{1,3}}{\|(x_k^{1,1},x_k^{1,2},x_k^{1,3})^T\|}\right)\end{bmatrix},\dots, \begin{bmatrix}\text{atan2}(x_k^{l,2},x_k^{l,1})\\ \arcsin\left(\frac{x_k^{l,3}}{\|(x_k^{l,1},x_k^{l,2},x_k^{l,3})^T\|}\right)\end{bmatrix}, \begin{bmatrix}\varphi_1\\ \theta_1\end{bmatrix},\dots,\begin{bmatrix}\varphi_d\\ \theta_d\end{bmatrix}\right),$$

$$f_k = \left(\left\| \begin{bmatrix} x_k^{1,1} \\ x_k^{1,2} \\ x_k^{1,3} \end{bmatrix} \right\|, \ldots, \left\| \begin{bmatrix} x_k^{l,1} \\ x_k^{l,2} \\ x_k^{l,3} \end{bmatrix} \right\|, x_k^{1,*}, \ldots, x_k^{d,*} \right),$$

where $(\alpha_1, \beta_1), \ldots, (\alpha_r, \beta_r)$ are the angles at which depth measurements are obtained. The expressions for p_k and f_k follow from the conversion of Cartesian into spherical coordinates as introduced in Sec. 3.2.

6. ENHANCEMENTS OF THE PROPOSED METHOD

In this section, we present several enhancements to the proposed method. These enhancements may not be absolutely necessary to apply the proposed methods but improve their performance and allow their use in a wider variety of applications.

6.1. Adaptive Addition of Nodes

When augmenting the state by adding control points, one has to choose the angles where the additional nodes should be located. In the 2D case, we need to determine the angles $\varphi_1, \ldots, \varphi_d$ and in the 3D case the pairs of angles $(\varphi_1, \theta_1), \ldots, (\varphi_d, \theta_d)$. Simple approaches may involve picking these angles at random within the view of the depth camera or choosing angles that lie on a grid.

However, these choices are usually not optimal. Additional control points should be added adaptively in areas where the error is large or where the expected gain in accuracy is high. One approach for the 2D case is to calculate the RMSE (root mean square error) between estimate and measurement

$$E_k^\tau(\alpha_i) = \sqrt{\frac{1}{\tau} \sum_{j=0}^{\tau-1} (s_{k-j}(\alpha_i) - \hat{z}_{k-j}^i)^2}$$

at time step k within a sliding window of length $\tau \in \mathbb{N}$ for each angle $\alpha_1, \ldots, \alpha_r$. A large RMSE suggests a control point at this position may be desirable and thus one should choose

$$\varphi = \underset{\alpha_i, 1 \le i \le r}{\arg\max} (E_k^\tau(\alpha_i))$$

as the angle of the new control point. This approach can be generalized to the 3D case by using the error function

$$E_k^\tau(\alpha_i, \beta_i) = \sqrt{\frac{1}{\tau} \sum_{j=0}^{\tau-1} (s_{k-j}(\alpha_i, \beta_i) - \hat{z}_{k-j}^i)^2}$$

and obtaining the new control point according to

$$(\varphi, \theta) = \underset{(\alpha_i, \beta_i), 1 \le i \le r}{\arg\max} (E_k^\tau(\alpha_i, \beta_i)).$$

In our experiments, we found that adding nodes successively tends to give better results than adding several nodes at once.

6.2. Handling Missing Measurements

In a practical setting, both position and depth measurements may be missing, for example when a tracked landmark is occluded or when the depth sensor is unable to provide a valid depth measurement at a certain angle. It is possible to handle these cases with slight modifications to the proposed method. The measurement models for both position and depth can simply omit the entries of $\underline{\hat{y}}_k$ and $\underline{\hat{z}}_k$ that could not be measured at time step k. Consequently, even a surface that is never visible as a whole at any given time step can be reconstructed over time.

6.3. Angular Uncertainty

The approach, as introduced before, assumes that the measurement angles $\alpha_1, \ldots, \alpha_r$ are exactly known and that only the depth values at these angles are affected by noise. This is a realistic assumption for certain depth sensors like TOF-cameras, where depth measurements are noisy but the measurement angles are fixed by the optical properties of the camera. However, depth sensors based on other measurement principles such as sensors based on structured light may not fulfill this assumption.

It is possible to extend the presented approach to include uncertainty in the measurement angles in order to more accurately model depth sensors whose measurement angles are affected by non-negligible noise. For this purpose, the measurement equation for depth has to be modified to include non-additive noise terms. In the 2D case, this yields

$$\underline{\hat{z}}_k = \begin{bmatrix} \hat{z}_k^1 \\ \vdots \\ \hat{z}_k^r \end{bmatrix} = \begin{bmatrix} s_k(\alpha_1 + \delta_k^1) \\ \vdots \\ s_k(\alpha_r + \delta_k^r) \end{bmatrix} + \underline{w}_k,$$

where $[\delta_k^1, \ldots, \delta_k^r]^T \sim \mathcal{N}(\underline{0}, \mathbf{C}_k^\delta)$ is zero-mean Gaussian noise with covariance $\mathbf{C}_k^\delta$. Nonlinear filters such as the UKF can handle this kind of noise, typically by means of augmenting the state with the non-additive noise components. This method is also applicable to the 3D case by changing the measurement equation to

$$\underline{\hat{z}}_k = \begin{bmatrix} \hat{z}_k^1 \\ \vdots \\ \hat{z}_k^r \end{bmatrix} = \begin{bmatrix} s_k(\alpha_1 + \delta_k^1, \beta_1 + \eta_k^1) \\ \vdots \\ s_k(\alpha_r + \delta_k^r, \beta_r + \eta_k^r) \end{bmatrix} + \underline{w}_k,$$

where $[\delta_k^1, \ldots, \delta_k^r, \eta_k^1, \ldots, \eta_k^r]^T \sim \mathcal{N}(\underline{0}, \mathbf{C}_k^{\delta,\eta})$ is zero-mean Gaussian noise with covariance $\mathbf{C}_k^{\delta,\eta}$.

6.4. More Than One Depth Camera

We previously assumed that our surface was observed by just a single depth camera. This assumption can be dropped by selecting a reference camera and transforming the depth measurements of additional cameras into the coordinate system of the reference camera. The relation between the coordinate systems

and the associated uncertainties can be obtained with standard camera calibration procedures [31], [10].

If the cameras are observing the surface from sufficiently different angles, angular uncertainty may be introduced by the transformation into a different coordinate system even if there was only uncertainty in depth initially. If necessary, the angular uncertainty can be handled as described in the previous section.

6.5. Choice of Interpolation or Approximation Function

In Sec. 3.1, we introduced the commonly used interpolation method based on thin plate splines. However, it should be noted that the proposed approach is not tied to a particular interpolation method. Other interpolation schemes can easily be used by replacing the interpolate$(p_k; f_k)$ function with a different algorithm. Overviews of possible interpolations methods can be found in [1], [33], and [21]. Depending on the application, the true shape of the surface may be more closely approximated by a certain type of interpolation. Furthermore, the choice of interpolation method influences the scalability of the algorithm for a large number of points m.

Furthermore, it is possible to consider a relaxed interpolation problem and to require approximation only. This means that $f(\underline{p}_i) = f_i$ does not need to hold exactly, but we only require that $f(\underline{p}_i) \approx f_i$. In the case of thin plate splines this can easily be achieved by setting $A(i,i) = \phi(0) + \lambda$ for $i = 1,\dots,m$ in Algorithm 1, where $\lambda > 0$ controls how strongly the problem is relaxed [27]. In our experiments, using an approximation rather than an interpolation has proven to be more stable, because outliers that result from poorly estimated points no longer affect the shape of the surface as much.

7. EVALUATION

In order to evaluate the proposed algorithm, we have performed several simulations. All simulations use the UKF [14] for nonlinear filtering. We use the following constants:

- initial estimate $\underline{x}_0^e$: uniformly random between $\underline{0}$ and $\underline{1}$
- initial covariance: $\mathbf{C}_0^e = 10 \cdot \mathbf{I}_{q\times q}$
- initial variance for additional nodes: $\sigma_{\text{add}}^2 = 10$
- noise covariance for position: $\mathbf{C}_k^v = 0.01 \cdot \mathbf{I}_{q\times q}$
- noise covariance for depth: $\mathbf{C}_k^w = 1 \cdot \mathbf{I}_{r\times r}$

For interpolation, we apply the RBF algorithm depicted in Algorithm 1 and use a scaled version of the thin plate spline as the RBF:

$$\phi(x) = \begin{cases} (x/1000)^2 \log(x/1000), & x > 0 \\ 0, & x = 0 \end{cases}.$$

7.1. Simulations in 2D

As a performance measure, we want to determine how similar the reconstructed surface is to the true surface. For this purpose, we choose $e \in \mathbb{N}$ evaluation angles $\gamma_1,\dots,\gamma_e$ and define the RMSE E_k of the estimated surface at time step k as

$$E_k = \sqrt{\frac{1}{e}\sum_{i=1}^{e}(s_k(\gamma_i) - s_k^{\text{true}}(\gamma_i))^2},$$

where $s_k^{\text{true}}(\cdot)$ is the true surface in polar coordinates. This can be interpreted as the error in depth, measured from the camera towards the surface.

We consider a depth camera with a viewing angle of 60° and a resolution of $r = 25$ measurements at equidistant measurement angles. There are $e = 26$ evaluation angles, which are equidistant in a 72° angle around the camera center, so we evaluate the extrapolation capability of the algorithm as well.

7.1.1 Static Case:

The true surface that we try to estimate is given by

$$s_k^{\text{true}}(\gamma) = 11 + 2\cos(9 \cdot \gamma)$$

and does not change over time. Thus, we omit the prediction step. This surface is the same as depicted in Fig. 6. We start with $l = 4$ landmarks and no additional nodes. From time step $k = 10$ to time step $k = 20$ we add one node at each time step, so we have $d = 11$ additional nodes afterwards. The angles $\varphi_1,\dots,\varphi_d$ are chosen deterministically and are evenly distributed across the camera's view.

The simulation was carried out repeatedly and the median and mean RMSE of the results of 100 Monte Carlo runs are shown in Fig. 7a. As can clearly be seen, the error is very high until time step $k = 10$, because the surface description does not have a sufficient number of degrees of freedom. After all additional nodes have been inserted at $k = 20$, the estimate quickly converges to a point where it has a consistently low error.

7.1.2 Dynamic Case:

We consider the same situation as in the static case except for the fact that the surface is now time-variant. The moving surface is given by

$$s_k^{\text{true}}(\gamma) = 11 + 2\cos(9 \cdot \gamma) + \sin(0.1 \cdot k)$$

and the system model is assumed to be unknown. Consequently, we use a random walk model for prediction. The system noise is modeled by the covariance matrix $\mathbf{C}_k^{\xi} = \text{diag}(0.1,\dots,0.1)$.

Once again, we performed 100 Monte Carlo runs and calculated the mean and median RMSE. The results are depicted in Fig. 7b. Overall the results look similar to the static case, but the RMSE is generally higher as is to be expected. The deviation between mean and median shows that there are a few outliers, so estimation is not quite as robust as in the static case.

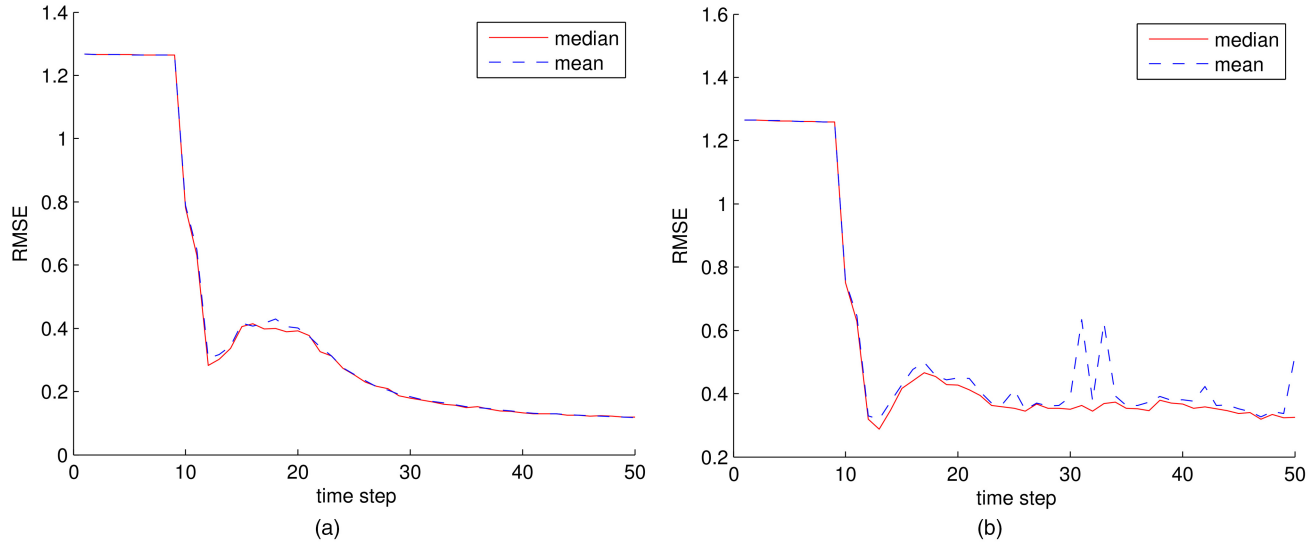


Fig. 7. Median and mean RMSE for each time step in the 2D case. Additional control points are inserted from time step $k = 10$ until $k = 20$. (a) Static surface. (b) Dynamic surface.

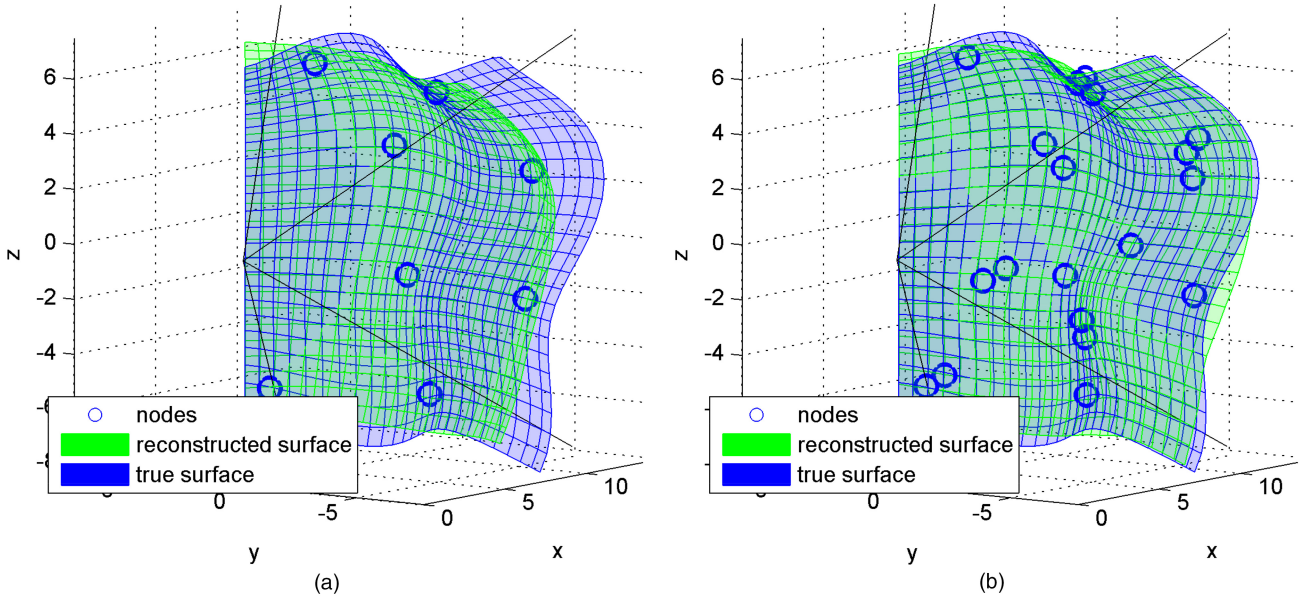


Fig. 8. Reconstructed and true surfaces in 3D (static case). (a) Time step $k = 9$, without additional control points. (b) Time step $k = 50$, with additional control points.

7.2. Simulations in 3D

Similar to the 2D case, we choose pairs of evaluation angles $(\gamma_1,\delta_1),\ldots,(\gamma_e,\delta_e)$ and define the RMSE E_k at time step k as

$$E_k = \sqrt{\frac{1}{e}\sum_{i=1}^{e}(s_k(\gamma_i,\delta_i) - s_k^{\text{true}}(\gamma_i,\delta_i))^2},$$

where $s_k^{\text{true}}(\cdot,\cdot)$ is the true surface in spherical coordinates.

We assume a depth camera with a horizontal and vertical viewing angle of 60° and a resolution of 25×25, so $r = 25^2 = 625$. The measurement angles are located on an equidistant 25×25 grid. For evaluation, we use vertical and horizontal angle of 72° and 26×26 equidistant evaluation angles.

7.2.1 Static Case:

Similar to the static 2D case, we consider a time-invariant surface. In spherical coordinates, it is given by

$$s_k^{\text{true}}(\gamma,\delta) = 12 + \sin(7\cdot\gamma) + \sin(7\cdot\delta).$$

Because the 3D surface has more degrees of freedom, we start with $l = 8$ landmarks. Once again, we introduce $d = 11$ additional control points from time step $k = 10$ to $k = 20$. The pairs of angles $(\varphi_1,\theta_1),\ldots,(\varphi_d,\theta_d)$ are evenly distributed across the field of view of the camera in a deterministic way. The surface before and after introducing additional nodes is depicted in Fig. 8. The

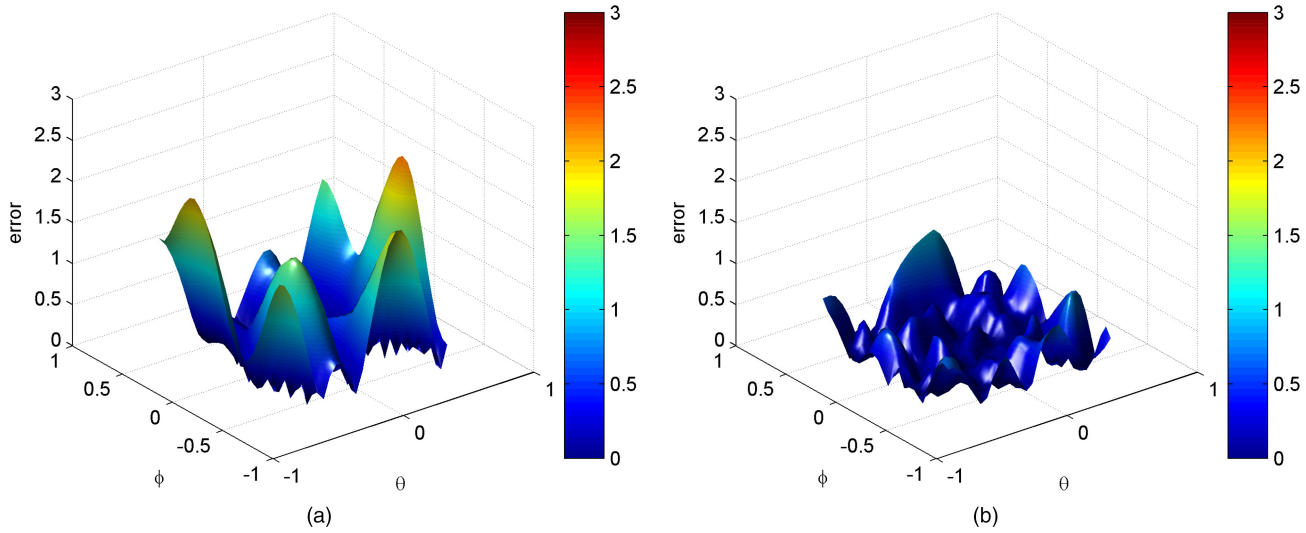


Fig. 9. Absolute error in 3D (static case). (a) Time step $k = 9$, without additional control points. (b) Time step $k = 50$, with additional control points.

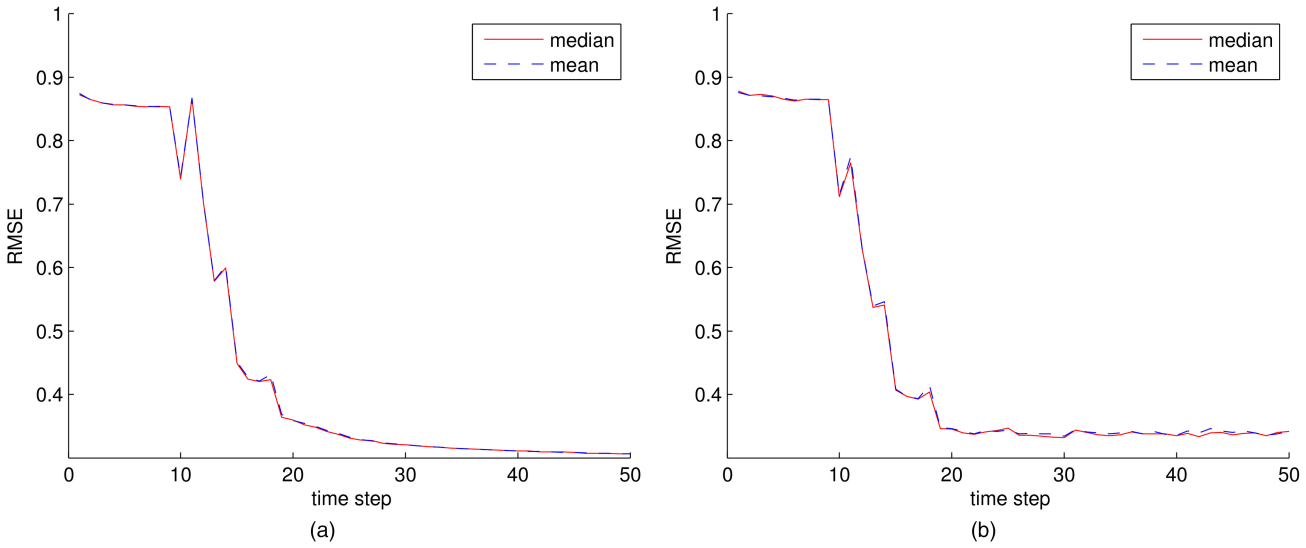


Fig. 10. Median and mean RMSE for each time step in the 3D case. Additional control points are inserted from time step $k = 10$ until $k = 20$. (a) Static case. (b) Dynamic case.

error between true and reconstructed surface can be seen in Fig. 9.

Our results from 100 Monte Carlo runs are shown in Fig. 10a. A comparison with Fig. 7a shows little difference to the 2D case, although the range of values is different because a different surface is reconstructed.

7.2.2 Dynamic Case:

For the dynamic 3D case we consider the time-variant surface

$$s_k^{\text{true}}(\gamma, \delta) = 12 + \sin(7 \cdot \gamma) + \sin(7 \cdot \delta) + \sin(0.1 \cdot k).$$

The system model is a random walk model with system noise $\mathbf{C}_k^c = \text{diag}(0.1, \ldots, 0.1)$. Fig. 10b shows the results from 100 Monte Carlo runs. This simulation demonstrates that our methods works almost as well in a dynamic as in a static setting.

7.3. Adaptive Addition of Nodes

In this section, we evaluate the approach proposed in Sec. 6.1 to add nodes adaptively in areas where the modeling error is large. We consider the same scenario as in the 2D static case (see Sec. 7.1.1). The sliding window has a length of $\tau = 9$. Three additional nodes are inserted at time steps $k = 10$, $k = 20$ and $k = 30$. For this simulation, we used an approximation with relaxation value $\lambda = 10^{-8}$ rather than an interpolation.

We performed 1000 Monte Carlo runs. The mean and median RMSE is depicted in Fig. 11a. The increase in accuracy is clearly visible each time a node is added. Fig. 11b shows the angles, at which the additional nodes are inserted. Obviously, these angles are not identical in each run, but it is clearly visible that the three additional nodes are mostly inserted at similar positions (approximately $-20°$, $0°$, and $20°$).

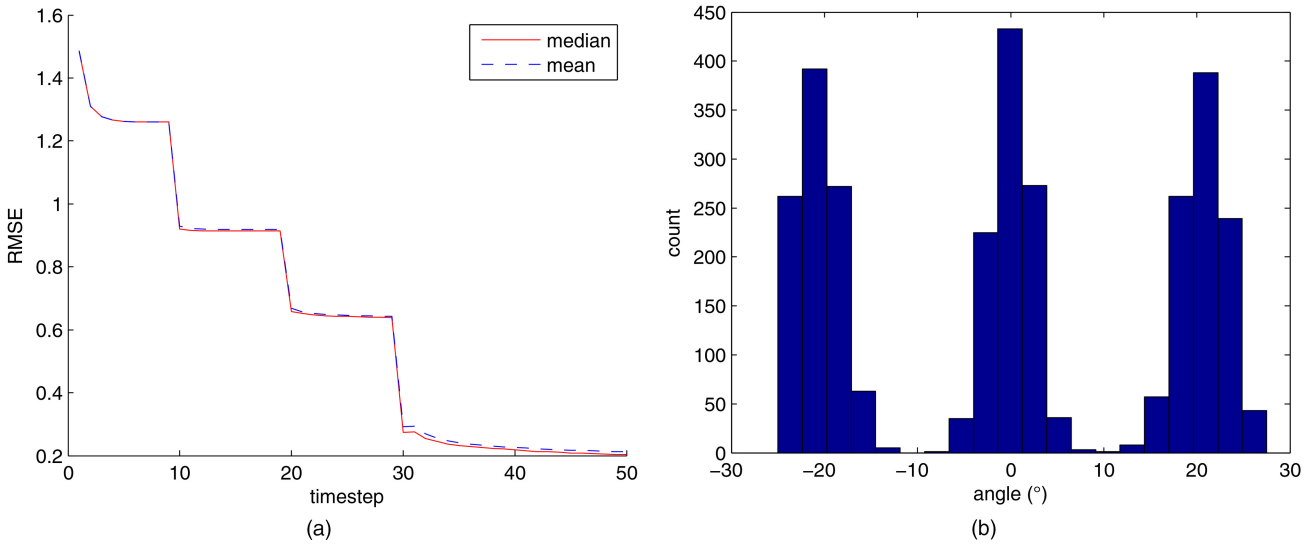


Fig. 11. Adaptive addition of nodes. Nodes are added at time steps $k = 10$, $k = 20$, and $k = 30$. (a) RMSE over time. (b) Angles where nodes are added.

8. CONCLUSION

We have presented an algorithm for recursively combining depth and position measurements for surface reconstruction considering uncertainties. Surface representation as a spline allows for a compact state representation. The measurement equation for position is trivial but deriving the measurement equation is challenging if Cartesian coordinates are used. In order to solve this issue, we have presented a way to use polar or spherical coordinates, which simplifies the problem significantly.

Through evaluation by means of simulations we have shown the viability of our approach in both 2D and 3D settings for static as well as dynamic surfaces. Our experiments clearly demonstrate the benefits of adding additional control points in order to better incorporate depth measurements. We also demonstrate the viability of the proposed method to adaptively choosing the location of additional nodes. The considered examples show how a much more accurate surface estimate can be obtained by the combined use of position as well as depth measurements.

Future research may include more sophisticated ways to insert additional control points. A practical application of the presented algorithm in a medical setting is planned.

ACKNOWLEDGMENT

This work was partially supported by the German Research Foundation (DFG) within the Research Training Groups RTG 1126 "*Intelligent Surgery—Development of new computer-based methods for the future working environment in visceral surgery*."

REFERENCES

[1] Amidror, I.
Scattered data interpolation methods for electronic imaging systems: A survey.
Journal of Electronic Imaging 11, 2 (Apr. 2002), 157–176.

[2] Arad, N., and Reisfeld, D.
Image warping using few anchor points and radial funcions.
Computer Graphics Forum 14 (Feb. 1995), 35–46.

[3] Baum, M., and Hanebeck, U. D.
Shape Tracking of Extended Objects and Group Targets with Star-Convex RHMs.
In *Proceedings of the 14th International Conference on Information Fusion (Fusion 2011)* (Chicago, Illinois, USA, July 2011).

[4] Besl, P., and McKay, N. D.
A method for registration of 3-D shapes.
IEEE Transactions on Pattern Analysis and Machine Intelligence 14, 2 (1992), 239–256.

[5] Brunn, D., and Hanebeck, U. D.
A Model-Based Framework for Optimal Measurements in Machine Tool Calibration.
In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA 2005)* (Barcelona, Spain, Apr. 2005).

[6] Carr, J. C., Fright, W. R., and Beatson, R. K.
Surface interpolation with radial basis functions for medical imaging.
IEEE Transactions on Medical Imaging 16 (1997), 96–107.

[7] Groch, A., Haase, S., Wagner, M., Kilgus, T., Kenngott, H., Schlemmer, H., Hornegger, J., Meinzer, H., and Maier-Hein, L.
A probabilistic approach to fusion of time of flight and multiple view based 3d surface reconstruction for laparoscopic interventions.
Int. J. CARS 7 (Suppl. 1) (2012), 397–398.

[8] Guan, L., Franco, J.-S., and Pollefeys, M.
3d object reconstruction with heterogeneous sensor data.
In *International Symposium on 3D Data Processing, Visualization and Transmission* (Atlanta, États-Unis, 2008).

[9] Gudmundsson, S. A., Aanaes, H., and Larsen, R.
Fusion of stereo vision and time-of-flight imaging for improved 3d estimation.
International Journal of Intelligent Systems Technologies and Applications 5, 3 (2008), 425–433.

[10] Hartley, R. I., and Zisserman, A.
Multiple View Geometry in Computer Vision, second ed.
Cambridge University Press, 2004.

[11] Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., and Stuetzle, W.
Surface reconstruction from unorganized points.
SIGGRAPH Comput. Graph. 26, 2 (July 1992), 71–78.

[12] Huber, M. F., and Hanebeck, U. D.
Gaussian Filter based on Deterministic Sampling for High Quality Nonlinear Estimation.
In *Proceedings of the 17th IFAC World Congress (IFAC 2008)* (Seoul, Republic of Korea, July 2008), vol. 17.

[13] Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A., and Fitzgibbon, A.
Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera.
In *ACM Symposium on User Interface Software and Technology* (Oct. 2011), pp. 559–568.

[14] Julier, S. J., and Uhlmann, J. K.
Unscented filtering and nonlinear estimation.
Proceedings of the IEEE 92, 3 (Mar. 2004), 401–422.

[15] Kalman, R. E.
A new approach to linear filtering and prediction problems.
Transactions of the ASME Journal of Basic Engineering 82 (1960), 35–45.

[16] Keeling, W. B., Williams, M. L., Slaughter, M. S., Zhao, Y., and Puskas, J. D.
Off-pump and on-pump coronary revascularization in patients with low ejection fraction: a report from the society of thoracic surgeons national database.
The Annals of Thoracic Surgery 96, 1 (July 2013), 83–89.

[17] Kurz, G., and Hanebeck, U. D.
Image Stabilization with Model-Based Tracking for Beating Heart Surgery.
In *11. Jahrestagung der Deutschen Gesellschaft für Computer- und Roboterassistierte Chirurgie (CURAC12)* (Düsseldorf, Germany, Nov. 2012).

[18] Kurz, G., and Hanebeck, U. D.
Recursive Fusion of Noisy Depth and Position Measurements for Surface Reconstruction.
In *Proceedings of the 16th International Conference on Information Fusion (Fusion 2013)* (Istanbul, Turkey, July 2013).

[19] Lee, S., Wolberg, G., and Shin, S. Y.
Scattered data interpolation with multilevel B-splines.
IEEE Transactions on Visualization and Computer Graphics 3, 3 (1997), 228–244.

[20] Lindner, M., Kolb, A., and Hartmann, K.
Data-fusion of pmd-based distance-information and high-resolution rgb-images.
In *International Symposium on Signals, Circuits and Systems, 2007.* (2007), vol. 1.

[21] Liu, W., and Ribeiro, E.
A survey on image-based continuum-body motion estimation.
Image and Vision Computing 29, 8 (2011), 509–523.

[22] Lorensen, W. E., and Cline, H. E.
Marching cubes: A high resolution 3d surface construction algorithm.
SIGGRAPH Comput. Graph. 21, 4 (Aug. 1987), 163–169.

[23] Marton, Z. C., Rusu, R. B., and Beetz, M.
On fast surface reconstruction methods for large and noisy point clouds.
In *Proceedings of the 2009 IEEE International Conference on Robotics and Automation (ICRA 2009)* (May 2009), pp. 3218–3223.

[24] Nakamura, Y., Kishi, K., and Kawakami, H.
Heartbeat synchronization for robotic cardiac surgery.
In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2001)* (Seoul, Korea, May 2001), pp. 2014–2019.

[25] Newcombe, R. A., Davison, A. J., Izadi, S., Kohli, P., Hilliges, O., Shotton, J., Molyneaux, D., Hodges, S., Kim, D., and Fitzgibbon, A.
Kinectfusion: Real-time dense surface mapping and tracking.
In *10th IEEE international symposium on mixed and augmented reality (ISMAR)* (Oct. 2011), IEEE, pp. 127–136.

[26] Rasmussen, C. E., and Williams, C. K. I.
Gaussian Processes for Machine Learning.
The MIT Press, 2006.

[27] Sprengel, R., Rohr, K., and Stiehl, H. S.
Thin-plate spline approximation for image registration.
In *Proceedings of the 18th Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (1996), vol. 3, IEEE, pp. 1190–1191.

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

[29] Stoyanov, D., Mylonas, G. P., Deligianni, F., Darzi, A., and Yang, G. Z.
Soft-tissue motion tracking and structure estimation for robotic assisted mis procedures.
In *Proceedings of the Medical Image Computing and Computer Assisted Interventions (MICCAI 2005)* (2005), vol. 2, pp. 139–146.

[30] Ye, C., and Borenstein, J.
Characterization of a 2D laser scanner for mobile robot obstacle negotiation.
In *Proceedings of the 2002 IEEE International Conference on Robotics and Automation (ICRA 2002)* (2002), vol. 3, pp. 2512–2518.

[31] Zhang, Z.
A flexible new technique for camera calibration.
IEEE Transactions on Pattern Analysis and Machine Intelligence 22, 11 (2000), 1330–1334.

[32] Zhu, J., Wang, L., Yang, R., Davis, J., and Pan, Z.
Reliability fusion of time-of-flight depth and stereo geometry for high quality depth maps.
IEEE Transactions on Pattern Analysis and Machine Intelligence 33, 7 (July 2011), 1400–1414.

[33] Zitová, B., and Flusser, J.
Image registration methods: a survey.
Image and Vision Computing 21 (2003), 977–1000.

Gerhard Kurz received his Dipl.-Inform. in computer science from the the Karlsruhe Institute of Technology (KIT), Germany, in 2012. Currently, he is working towards a Ph.D. degree at the Intelligent Sensor-Actuator-Systems Laboratory, Karlsruhe Institute of Technology (KIT), Germany. His research interests are in the field of medical data fusion, nonlinear estimation, and directional filtering.

Uwe D. Hanebeck received his Dipl.-Ing. in Electrical Engineering, with first-class honors, from the Ruhr-Universitaet Bochum, Germany, in 1991. He obtained his Dr.-Ing. (Ph.D.) degree in Electrical Engineering from the Technische Universitaet Muenchen, Germany, in 1997. In 2003 he was appointed as professor at the Universitaet Karlsruhe (TH). He is the head of the Intelligent Sensor-Actuator-Systems Laboratory (ISAS), Karlsruhe Institute of Technology (KIT). His research interests are sensor-actuator-networks, human-centered robotics, and telepresence with emphasis on theory-driven approaches based on stochastic estimation theory and uncertainty models. Research results are applied to various fields like human-robot-cooperation, robotic surgery systems, personal assistance systems, and extended-range telepresence.

The PMHT for Passive Radar in a DAB/DVB Network

SORA CHOI
PETER WILLETT
SHENGLI ZHOU

Passive radar using Digital Audio/Video Broadcast (DAB/DVB) signals with Orthogonal Frequency Division Multiplexing (OFDM) must contend with measurements of range and range-rate only (no, or very poor, angular information) and must deal with an added and unwonted measurement-illuminator-target association. But tracking systems using modified Joint Probabilistic Data Association (JPDA) filters and using particle filters have been suggested and seem to work effectively to maintain tracks directly in the Cartesian domain. In this correspondence, we present an alternative Cartesian-domain tracking algorithm a version of the Probabilistic Multi-Hypothesis Tracker (PMHT), to contend with the extra-list data association in a natural way.

Manuscript received December 6, 2013; revised March 7, 2014; released for publication March 7, 2014.

Refereeing of this contribution was handled by Huimin Chen.

This work was supported by the Office of Naval Research under contracts N00014-09-10613, N00014-10-10412 & N000014-13-1-0231.

Authors' address: Department of Electrical and Computer Engineering, University of Connecticut, 371 Fairfield Way, U-4157, Storrs, Connecticut 06269 USA, (E-mail: {soc07002, willett, shengli}@engr.uconn.edu).

I. INTRODUCTION

Passive Bistatic Radar (PBR), also known as Passive Coherent Location (PCL), uses illuminators of opportunity. Passive radar using signals in a single frequency network modulated according to the Digital Audio/Video Broadcasting (DAB/DVB) standards using orthogonal frequency division multiplexing (OFDM) has recently been of increasing interest. There has been considerable research to develop tracking systems addressing its inherent difficulties [3]–[7], [10]–[13]; the poor quality—or absence—of angular information, and the lack of label of the transmitter on top of the usual target/measurement association concerns.

First, there are algorithms using the Multi-Hypothesis Tracker (MHT) [12], [13] addressing the complexity problem from association ambiguities between measurement, targets and illuminators by initially forming two dimensional (measurement-target) hypotheses in the two-dimensional range/Doppler domain. Tracking is thence performed directly on target parameters by the MHT without considering the association between measurements and illuminators: the range/Doppler MHT extracts measurements and removes false alarms. Then, de-ghosting is performed by evaluating likelihood probabilities of possible data associations. When a Cartesian track is confirmed, the remaining tracks from other possible associations are declared false and tracking starts in the Cartesian domain.

This MHT approach is good but but is not without issues. One is the appropriate motion model in range and Doppler space: probably the target dynamics in the Cartesian domain are known, the trajectories are not easily described in a space of target parameters, because the trajectories are related to illuminator/receiver/target geometry and there is association ambiguity among measurements, illuminators, and targets. And that is another concern: the illuminator association is never explicitly addressed.

Now, track maintenance algorithms that operate directly in Cartesian coordinates have been explored [4], [5], one using modified Joint Probabilistic Data Association (JPDA) and another a particle filter. For the former, in order to address the large number of three-list hypotheses, a "super-target" idea was proposed; and the particle filters work under the PMHT measurement model that each measurement's assignments are independent of others'. These methods have also been examined downstream from an initiation approach (the PMHTI method, suggested in [6]) that initiates tracks in Cartesian coordinates.

In fact the PMHT seems to be an effective and natural way to accommodate the data association with the extra list (transmitters). So in this paper, we present it: it is really very simple. This tracker, combined with the initiation algorithm (the modified PMHTI method in [6]), shows excellent performance in comparison with the JPDA filter and particle filter.

Section II explains what tracking in the Cartesian domain involves. Section III presents the PMHT solution, as well as a brief summary of the modified initiation algorithm. There are results in Section IV. We wish to note that although the motivating example is DAB/DVB passive radar, the techniques here could apply to any multistatic system (e.g., sonar) with common transmitter waveform. The common key ingredient is the measurements' lack of illuminator label.

II. MODEL

A. Process

We assume that there are multiple targets. For the mth target the state

$$\mathbf{x}_m(t_i) = [x_m(t_i), \dot{x}_m(t_i), y_m(t_i), \dot{y}_m(t_i), z_m(t_i), \dot{z}_m(t_i)]$$

is to be estimated, and comprises its location $\mathbf{p}_m(t_i) = [x_m(t_i), y_m(t_i), z_m(t_i)]^T$ and velocity $\mathbf{v}_m(t_i) = [\dot{x}_m(t_i), \dot{y}_m(t_i), \dot{z}_m(t_i)]^T$.

Each target moves according to a model that makes sense in the Cartesian[1] domain, such as according to kinematic dynamics or constant-speed turn. We assume there are M_{t_i} targets at time t_i, such that the goal is to estimate $\{\{\mathbf{x}_m(t_i)\}_{m=1}^{M_{t_i}}\}_{i=1}^{T}$. Track management (determination of M_{t_i}) is not the subject of this correspondence; however, we suggest the techniques of [6] for initiation (the more difficult component) and we later give some suggestions for termination. We do not address track-merging or -spawn.

There are N_s illuminators, the sth being located at $\mathbf{x}_s = [x_s, y_s, z_s]^T$; and there is assumed a single receiver at $\mathbf{x}_r = [x_r, y_r, z_r]^T$. The receiver can for target m measure bistatic range $\gamma(t_i)$ and range-rate $\dot{\gamma}(t_i)$, which are given by

$$\gamma(\mathbf{x}_m(t_i), \mathbf{x}_s) = \|\mathbf{p}_m(t_i) - \mathbf{x}_r\| + \|\mathbf{p}_m(t_i) - \mathbf{x}_s\| \tag{1}$$

$$\dot{\gamma}(\mathbf{x}_m(t_i), \mathbf{x}_s) = \frac{(\mathbf{p}_m(t_i) - \mathbf{x}_r)^T \cdot \mathbf{v}_m(t_i)}{\|\mathbf{p}_m(t_i) - \mathbf{x}_r\|} + \frac{(\mathbf{p}_m(t_i) - \mathbf{x}_s)^T \cdot \mathbf{v}_m(t_i)}{\|\mathbf{p}_m(t_i) - \mathbf{x}_s\|} \tag{2}$$

in which $s \in \{1, \ldots, N_s\}$. The generic observation vector in the absence of noise is given by

$$h_s(\mathbf{x}(t_i)) = (\gamma(\mathbf{x}(t_i), \mathbf{x}_s), \dot{\gamma}(\mathbf{x}(t_i), \mathbf{x}_s))^T. \tag{3}$$

Hence, the measurement at time t_i for target m and involving illuminator s is

$$z(t_i) = h_s(\mathbf{x}_m(t_i)) + \nu_{m,s}(t_i) \tag{4}$$

with $\nu_{m,s}(t_i)$ independent, zero-mean and Gaussian measurement noises of covariance $R_{m,s}(t_i)$.

While (4) is a function of the target $\mathbf{x}_m(t_i)$ and transmitter $\mathbf{x}_s$, the observations available to the tracker at time t consist of a set $\{z^{(r)}(t_i)\}$ with components unlabeled as to m nor s—the uncertainty as to the transmitter s is a new visitor to the usual "measurement-origin uncertainty" (MOU) model. On the other hand, a familiar ingredient to MOU is that although all combinations of s and m (that is: $N_s \times M$ elements) might be thought an "original" set of measurements, but these are thinned according to a Bernoulli process: each is retained with probability P_d and else discarded. Another familiar ingredient is that the set of survivors be augmented by a Poisson set of false alarms uniformly distributed in range & range-rate.

The PMHT model [8], [16], [19] is different, and will be described in detail in the next section. However, its features are:

- that the model is not *generative*: that is, it is posterior to knowing the number of measurements available;
- that the a priori association probabilities (of each measurement having arisen from target m and due to transmitter s) are independent for each measurement; and
- that each target/transmitter pair be represented at most once per time t_i in the observation set is not enforced;

The model is not reflective of our passive-radar physics; however, it is clear and unashamed, and it results in a feasible algorithm that works quite effectively.

III. THE PMHT

A. Description of the algorithm

When there are N_s illuminators[2] and M targets, at each measurement time t_i ($i = 1, 2, \ldots, T$), a state of target m is denoted by $\mathbf{x}_m(t_i)$. X_T is the collection of states for all targets up to time T and Z_T is the set of conditionally statistically independent measurements, in which $Z_T = (Z_{t_0}, Z_{t_1}, \ldots, Z_{t_T})$, where $Z_{t_i} = (\mathbf{z}_{t_i}^{(1)}, \mathbf{z}_{t_i}^{(2)}, \ldots, \mathbf{z}_{t_i}^{(N_{t_i})})$—there are N_{t_i} measurements at time t_i. In fact, we have:

- The statistically independent measurement-to-target assignments are described by $\mathcal{K}_T = (\mathcal{K}_{t_0}, \mathcal{K}_{t_1}, \ldots, \mathcal{K}_{t_T})$ and $\mathcal{K}_{t_i} = (k_{t_i}^{(1)}, k_{t_i}^{(2)}, \ldots, k_{t_i}^{(N_{t_i})})$ where $1 \le k_{t_i}^{(r)} \le M$ denotes the index of the target assigned to the measurement $z_{t_i}^{(r)}$.
- The (again: statistically independent) measurement-to-illuminator assignments—the new illuminator-to-measurement association ambiguity—are denoted $\mathcal{L}_T = (\mathcal{L}_{t_0}, \mathcal{L}_{t_1}, \ldots, \mathcal{L}_{t_T})$, in which $\mathcal{L}_{t_i} = (l_{t_i}^{(1)}, l_{t_i}^{(2)}, \ldots, l_{t_i}^{(N_{t_i})})$ and $1 \le l_{t_i}^{(r)} \le N_s$ is the index of illuminator assigned to the measurement $\mathbf{z}_{t_i}^{(r)}$.

The "usual" PMHT association model accommodates more than one measurement being assigned to each target, and that remains so here. But note also that each target can further be associated with more than one measurement for each illuminator. Physically this is wrong; algorithmically it is a nice feature.

[1] Since these models are well known, this short paper simply defers to references, for example [1].

[2] We use the subscript s mostly to avoid confusion, but defensibly to indicate the ground-station in the DVB/DAB network.

In the usual case having only one kind of data association ambiguities between measurements and targets, the PMHT uses the EM algorithm to maximize $p(X_T \mid Z_T)$ over X_T [8], [9], [19], utilizing the quantity

$$Q(X_T^{(n+1)} \mid X_T^{(n)}) \equiv \sum_{\mathcal{K}_T} \ln(p(X_T^{(n+1)}, \mathcal{K}_T \mid Z_T)) p(\mathcal{K}_T \mid X_T^{(n)}, Z_T), \quad (5)$$

instead of directly finding the MAP (maximum a posteriori) estimate of X_T

$$X_{\text{MAP}} = \arg\max_{X_T} E\{\ln(p(X_T \mid Z_T))\}. \quad (6)$$

At each iteration, the algorithm finds

$$X_T^{(n+1)} = \arg\max_{X_T^{(n+1)}} Q(X_T^{(n+1)} \mid X_T^{(n)}) \quad (7)$$

achieving $p(X_T^{(n+1)} \mid Z_T) > p(X_T^{(n)} \mid Z_T)$.

Generalizing this to the case having the additional association ambiguities between measurements and illuminators, the quantity $Q(X_T^{(n+1)} \mid X_T^{(n)})$ is

$$Q(X_T^{(n+1)} \mid X_T^{(n)}) \equiv \sum_{\mathcal{K}_T, \mathcal{L}_T} \ln(p(X_T^{(n+1)}, \mathcal{K}_T, \mathcal{L}_T \mid Z_T)) \times p(\mathcal{K}_T, \mathcal{L}_T \mid X_T^{(n)}, Z_T). \quad (8)$$

Assuming $P(k_{t_i}^{(r)} = m) = \pi_m^k$ and $P(l_{t_i}^{(r)} = s) = \pi_s^l$, the conditional pdf for all measurements is as follows:

$$P(\mathcal{K}_T, \mathcal{L}_T \mid X_T^{(n)}, Z_T) = \prod_{i=1}^{T} \prod_{r=1}^{N_{t_i}} \omega_{t_i,r}^{(n)}(m,s) \quad (9)$$

in which

$$\omega_{t_i,r}^{(n)}(m,s) = \frac{\pi_m^k \pi_s^l p(\mathbf{z}_{t_i}^{(r)} \mid k_{t_i}^{(r)} = m, l_{t_i}^{(r)} = s, \mathbf{x}_m^{(n)}(t_i))}{\sum_{p=1}^{M} \sum_{q=1}^{N_s} \pi_p^k \pi_q^l p(\mathbf{z}_{t_i}^{(r)} \mid k_{t_i}^{(r)} = p, l_{t_i}^{(r)} = q, \mathbf{x}_p^{(n)}(t_i))}. \quad (10)$$

It is noted that $\omega_{t_i,r}^{(n)}(m,s)$ denotes the posterior probability of measurement r being related to target m and illuminator s at time t_i at the nth EM iteration.

Now, we have

$$Q(X_T^{(n+1)} \mid X_T^{(n)}) = \sum_{\mathcal{K},\mathcal{L}} \ln[p(X^{(n+1)}, \mathcal{K}, \mathcal{L}, Z_T) p(\mathcal{K}, \mathcal{L} \mid X^{(n)}, Z_T)] \quad (11)$$

$$= \ln\left[\prod_{q=1}^{M} (p(\mathbf{x}_q^{(n+1)}(t_1)) \prod_{i=2}^{T} p(\mathbf{x}_q^{(n+1)}(t_i) \mid \mathbf{x}_q^{(n+1)}(t_{i-1}))\right] \quad (12)$$

$$+ \sum_{m=1}^{M_t} \sum_{s=1}^{N_s} \sum_{i=1}^{T} \sum_{r=1}^{N_{t_i}} (\omega_{t_i,r}^{(n)}(m,s) \ln[\pi_m^k \pi_s^l] + \omega_{t_i,r}^{(n)}(m,s) \ln[p(\mathbf{z}_{t_i}^{(r)} \mid k_{t_i}^{(r)} = m, l_{t_i}^{(r)} = s, \mathbf{x}_m^{(n+1)}(t_i))])$$

so

$$\nabla_{X^{(n+1)}} Q(X^{(n+1)} \mid X^{(n)}) \quad (13)$$

$$= \nabla_{X^{(n+1)}} \left[\ln\left[\prod_{q=1}^{M} (P(\mathbf{x}_q^{(n+1)}(t_1)) \times \prod_{i=2}^{T} P(\mathbf{x}_q^{(n+1)}(t_i) \mid \mathbf{x}_q^{(n+1)}(t_{i-1}))\right]\right] + \sum_{m,s} \sum_{i=1}^{T} \nabla_{X^{(n+1)}} h_s(\mathbf{x}_m^{(n+1)}(t_i)) \times \left[\left(\frac{R_{m,s}(t_i)}{\sum_{r=1}^{N_{t_i}} \omega_{t_i,r}^{(n)}(m,s)}\right)^{-1} \times \left(\sum_{r=1}^{N_{t_i}} \frac{\omega_{t_i,r}^{(n)}(m,s)(\mathbf{z}_{t_i}^{(r)})}{\sum_{r=1}^{N_{t_i}} \omega_{t_i,r}^{(n)}(m,s)} - h_s(\mathbf{x}_m^{(n+1)}(t_i))\right)\right]$$

Similar to [19], $\nabla_{X_T^{(n+1)}} Q(X_T^{(n+1)} \mid X_T^{(n)})$ is equal to the gradient of logarithm of the joint "synthetic" Q-function $\tilde{Q}(X_T^{(n+1)} \mid X_T^{(n)})$ having no data association uncertainty with synthetic measurement $\tilde{\mathbf{z}}_{m,s}$ and synthetic measurement covariance $\tilde{R}_{m,s}(t_i)$. That is, we have

$$\nabla_{X^{(n+1)}} Q(X_T^{(n+1)} \mid X_T^{(n)}) = \nabla_{X_T^{(n+1)}} \tilde{Q}(X_T^{(n+1)} \mid X_T^{(n)}) \quad (14)$$

where

$$\tilde{Q}(X_T^{(n+1)} \mid X_T^{(n)}) = \ln\left[\prod_{q=1}^{M} (p(\mathbf{x}_q^{(n+1)}(t_1)) \prod_{i=2}^{T} p(\mathbf{x}_q^{(n+1)}(t_i) \mid \mathbf{x}_q^{(n+1)}(t_i - 1))\right] - \frac{1}{2} \sum_{m=1}^{M_t} \sum_{s=1}^{N_s} \sum_{i=1}^{T} [\tilde{\mathbf{z}}_{t_i}(m,s) - h_s(\mathbf{x}_m^{(n+1)}(t_i))]^T \times \tilde{R}_{m,s}(t_i)^{-1} [\tilde{\mathbf{z}}_{t_i}(m,s) - h_s(\mathbf{x}_m^{(n+1)}(t_i))], \quad (15)$$

we obtain

$$\tilde{\mathbf{z}}_{m,s}(t_i) = \frac{\sum_{r=1}^{N_{t_i}} \omega_{t_i,r}^{(n)}(m,s) \mathbf{z}_{t_i}^{(r)}}{\sum_{r=1}^{N_{t_i}} \omega_{t_i,r}^{(n)}(m,s)} \quad (16)$$

$$\tilde{R}_{m,s}(t_i) = \frac{R_{m,s}(t_i)}{\sum_{r=1}^{N_{t_i}} \omega_{t_i,r}^{(n)}(m,s)} \quad (17)$$

The form of (15) reflects the nonlinearity of the measurement model (4) by involving $h_s(\mathbf{x}_m^{(n+1)}(t_i))$ explicitly. If one's taste leans towards the unscented Kalman filter (UKF) (e.g., [2]) this is useful. Perhaps it is more familiar to write

$$H_{m,s}(t_i) = \nabla_{X^{(n)}} h_s(\mathbf{x}_m^{(n)}(t_i)) \quad (18)$$

suggesting that an extended Kalman filter (EKF) [1] can be used within a Kalman smoother.

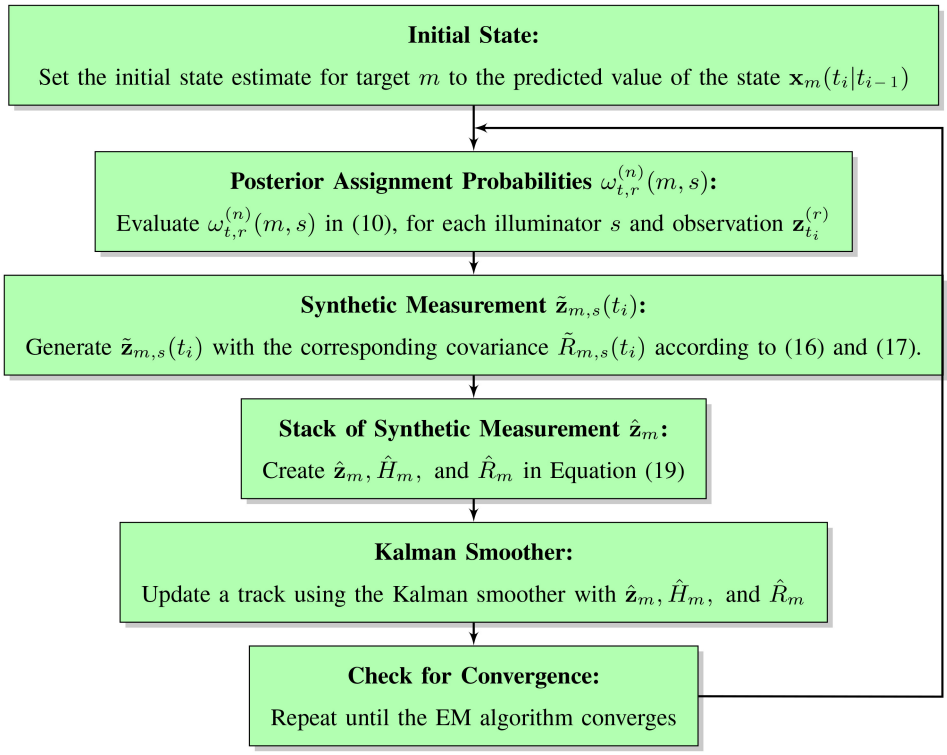


Fig. 1. PMHT (tracking) algorithm to update the state and covariance of target m at time t_i. If an EKF is used, $\tilde{\mathbf{z}}_{m,s}(t_i)$ in the third step should be replaced by $\tilde{\mathbf{z}}_{m,s}(t_i) - [h_s(\mathbf{x}_m^{(n)}(t_i)) - H_{m,s}(t_i)\mathbf{x}_m^{(n)}(t_i)]$.

That is, an iterated (extended) Kalman smoother routine using "synthetic" measurements $\{\tilde{\mathbf{z}}_{m,s}(t_i)\}_{i=1}^T$ and corresponding covariances $\{\tilde{R}_{m,s}(t_i)\}_{i=1}^T$ from (16) and (17) for each target m would be sufficient if there were only one transmitter $s = N_s = 1$, although a full Cartesian track would problematic if there were only one transmitter. To exploit the multi-sensor "triangulation" necessary for Cartesian tracking it is necessary to fuse data from multiple illuminators. There are various methods [8], [14], [15], [17] for this, and we adopt here that using stacked synthetic measurements:

$$\hat{\mathbf{z}}_m(t_i) = [\tilde{\mathbf{z}}_{m,1}^T(t_i), \tilde{\mathbf{z}}_{m,2}^T(t_i), \ldots, \tilde{\mathbf{z}}_{m,N_s}^T(t_i)]^T \tag{19}$$

$$\hat{H}_m(t_i) = \text{diag}[H_{m,1}(t_i), H_{m,2}(t_i), \ldots, H_{m,N_s}(t_i)]$$

$$\hat{R}_m(t_i) = \text{diag}[\tilde{R}_{m,1}(t_i), \tilde{R}_{m,2}(t_i), \ldots, \tilde{R}_{m,N_s}(t_i)]$$

The algorithm is described in Figure 1. We note that the PMHT is known sometimes to converge to a local MAP (basically a lost track) [19]. In our study here we have not suppressed this behavior; but it seems to be a lesser problem for the PMHT than the additional data association that vexes the other approaches.

In the following section we will compare this new method to the modified JPDA and particle filter (PF) approaches. Hence in the next two subsections we will mention the track-management schemes used for this comparison.

B. Initiation of tracks

To initiate tracks, we adopt the PMHTI method presented in [6]. To decrease complexity caused by the search for local maxima of $p(Z \mid \mathbf{p})$ using *all* the points obtained by the spherical-intersection method, we modify the search step to use only the initialization points having the highest likelihoods (rather than all the points). We call this the modified PMHTI method in Figure 2.

Here, a track is confirmed in the PMHT if the sum of measurement weights is higher than a certain threshold for three out of five consecutive scans, and the estimated covariance is smaller than a certain threshold. The JPDA filter confirms a track if there is at least one measurement within the target's gate for three out of five consecutive scans and the gate is smaller than a certain threshold. The particle filter confirms temporary tracks if there is at least one measurement in the validation region for three out of five consecutive scans.

C. Track Termination

Respectively, these use:

1) PMHT: In the PMHT, a track is called lost if the sum of weights of measurements (i.e., $\sum_s \omega_{t_i,r}^{(n)}(m,s)$) is less than a certain threshold for three out of five consecutive scans, or the estimated covariance is larger than a certain threshold.

2) JPDA using the Extend Kalman filter: The track is declared lost if no measurement falls within the

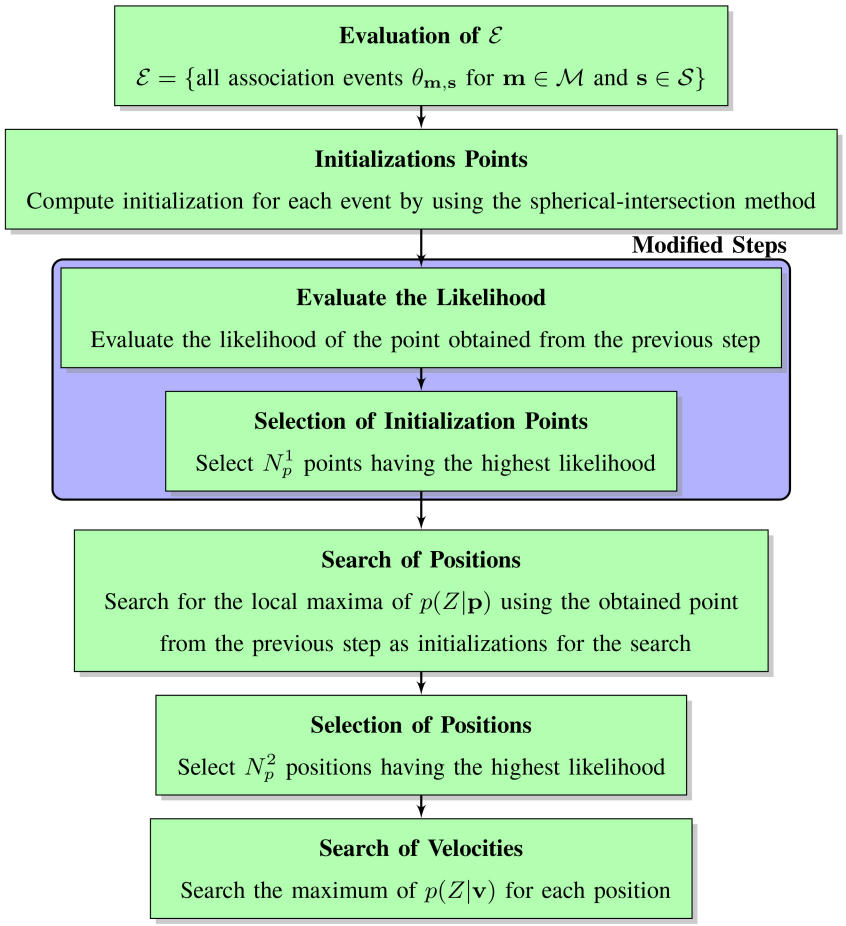


Fig. 2. Modified PMHTI (initiation) method: $\mathcal{M}$ is a set of 3-permutations of measurements and $\mathcal{S}$ is a set of 3-combinations of illuminators. N_p^1 and N_p^2 are thresholds, and the position and velocity is denoted by $\mathbf{p}$ and $\mathbf{v}$, respectively.

target's gate [2] for three out of five consecutive scans or the gate is larger than a certain threshold. When the predicted measurement $\hat{\mathbf{z}}$ and the associated covariance S are given, the measurement $\mathbf{z}$ is considered to be in the validation region if $(\mathbf{z}-\hat{\mathbf{z}})^T S^{-1}(\mathbf{z}-\hat{\mathbf{z}}) < \gamma$, where the threshold γ denotes the gate size.

3) PF: The particle filter defines the validation region by using statistical distance in measurement space [2], [18]: when $\hat{\mathbf{z}}$ is the converted measurement from an estimated state and R is the measurement noise covariance, the measurement $\mathbf{z}$ is defined as valid if $(\mathbf{z}-\hat{\mathbf{z}})^T R^{-1}(\mathbf{z}-\hat{\mathbf{z}}) < \gamma$ for threshold γ. If there is no measurement in the validation region for three out of five consecutive scans, or every particle has negligible weight caused by impoverishment/degeneracy of particles, the track is declared lost. It is noted that the particle filter does not use the validation region to estimate the state, unlike the JPDA.

IV. SIMULATION

It is assumed that there are three targets, five illuminators and one receiver, as in Figure 3. False measurements are uniformly distributed with spatial density λ in a surveillance region of volume V in range-/range-rate space, while their number ϕ is Poisson. The measurement noise follows a Gaussian distribution $\mathcal{N}(0,\sigma_\gamma^2)$ for range, and $\mathcal{N}(0,\sigma_{\dot{\gamma}}^2)$ for range-rate. 100 Monte Carlo runs are performed and 2,000 particles are used in the particle filter. Track management is integral to the simulation: no tracker has prior information as to the number of targets. For the initiation method $N_p^1 = 50$ and $N_p^2 = 3$ are chosen (see Figure 2). The PMHT uses a sliding batch (see [19]) of length 5.

A. Comparison in Terms of Target Number

We compare the PMHT to the modified JPDA filter and the (bootstrap) particle filter [5], where in each case the PMHTI method [6] is used for track initiation and using track confirmation/termination as described earlier. Trajectories from the previous simulation (pictured in Figure 3) are used, but they begin at different times and last for fewer than 60 scans in order to examine the performance of track management. In this subsection, the measurement noise standard deviation is $\sigma_\gamma = 20$ m and $\sigma_{\dot{\gamma}} = 2$ m/s.

The PMHT filter confirms and terminates tracks remarkably well, regardless of the false alarm density and the detection probability, as seen in Figures 4 and

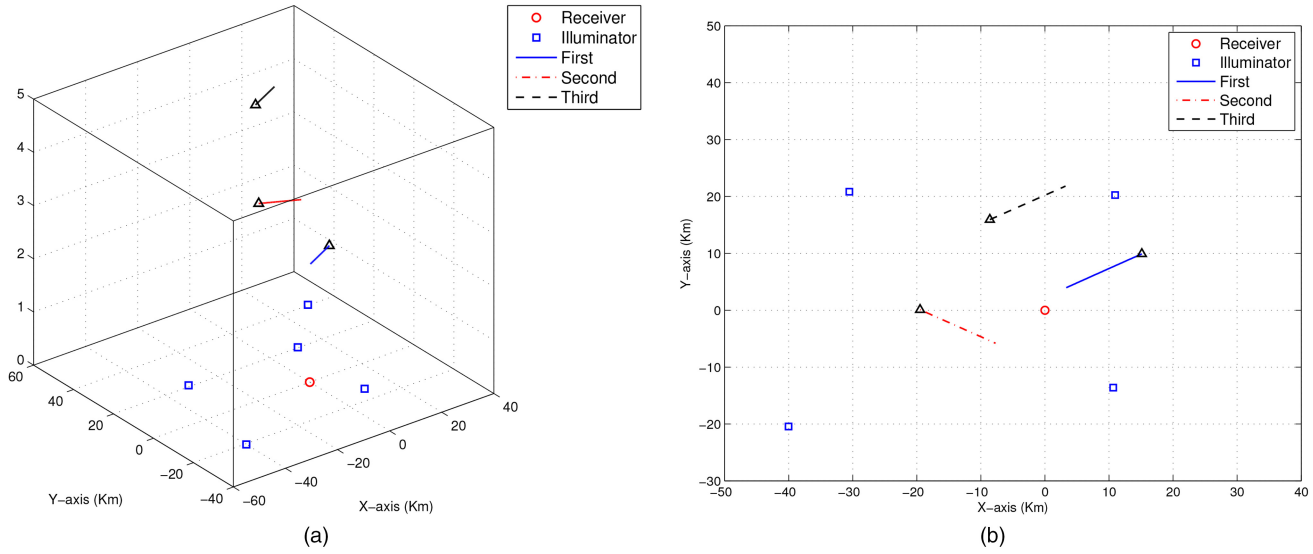


Fig. 3. Trajectories with three targets, five illuminators and one receiver. (a) Three dimensional space. (b) Projected in a two dimensional space.

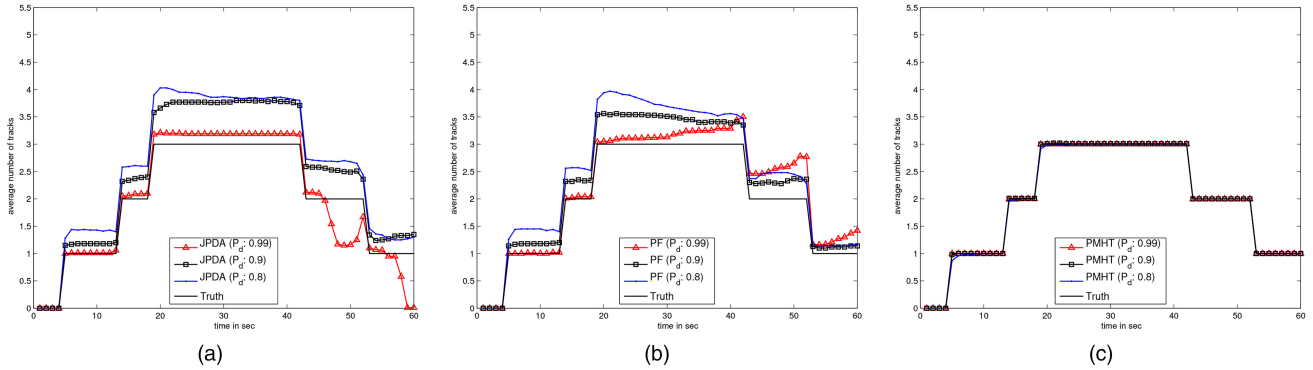


Fig. 4. Average number of confirmed tracks with low false alarm density ($\lambda V = 1$): the PMHT almost overlaps with the ground truth. (a) JPDAF. (b) Particle Filter. (c) PMHT.

6, where the performance almost does not degrade for the high false alarm density. Although the JPDA and particle filters show good performance with a low false alarm density, when the expected number of false alarms is increased to four the termination of tracks in the JPDA filter degrades (track confirmation is still quite good) and the particle filter deteriorates according to both measures.

As expected, the number of temporary tracks in every tracker increases at higher false alarm rates, as seen in Figures 5 and 7. It is also shown that this number is strongly related to the number of confirmed tracks. For example, since the particle filter does not perform termination and confirmation especially well in high clutter, there are more confirmed tracks and fewer temporary ones compared to the other trackers in Figures 6 and 7, since if more measurements are within the validation region of the confirmed tracks there are fewer measurements to initiate temporary tracks.

It is noted that the modifications to the termination rule affect track confirmation, for example a stricter termination rule could terminate temporary tracks too easily before they are confirmed. When stricter termination and confirmation rules are applied to these filters (e.g., two out of seven scans for termination and five out of seven scans for confirmation), the performance of the JPDA filter is severely degraded and the particle filter's performance is also worsened, especially at lower detection probabilities. In the PMHT filter, the threshold on the sum of the weights is more influential on the performance than the choice of termination/confirmation rules and the appropriate threshold is also critical to show the good performance.

We also note that in this simulation the PMHT, with its simplified data association, does not require many iterations to converge; while the JPDAF still needs to evaluate all data association events and the particle filter must update every particle's weight. Hence, at least in our experience, the PMHT is considerably faster than the JPDA and particle filter approaches.

B. Comparison in Terms of Track Accuracy

We have carried out additional comparison of the trackers via RMSE. In order to remove the effect of

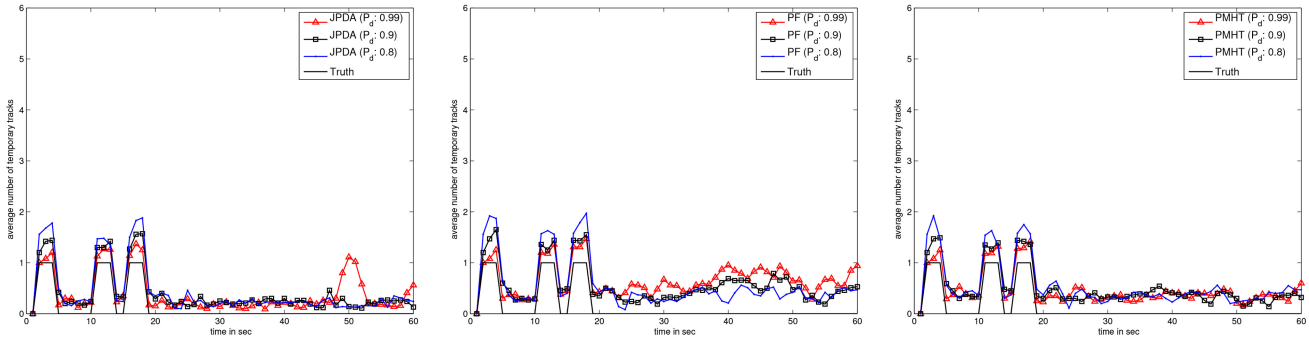


Fig. 5. Average number of temporary tracks with the low false alarm density ($\lambda V = 1$): the truth is the number of true tracks considered as temporary tracks following the track confirmation rule. (a) JPDAF. (b) Particle Filter. (c) PMHT.

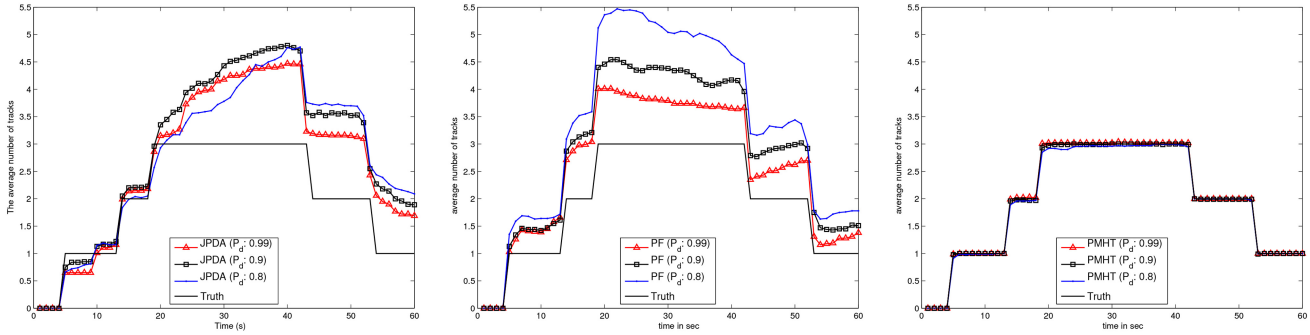


Fig. 6. Average number of confirmed tracks with high false alarm density ($\lambda V = 4$): the PMHT shows similar performance to the case having low false alarm density, unlike other filters. (a) JPDAF. (b) Particle Filter. (c) PMHT.

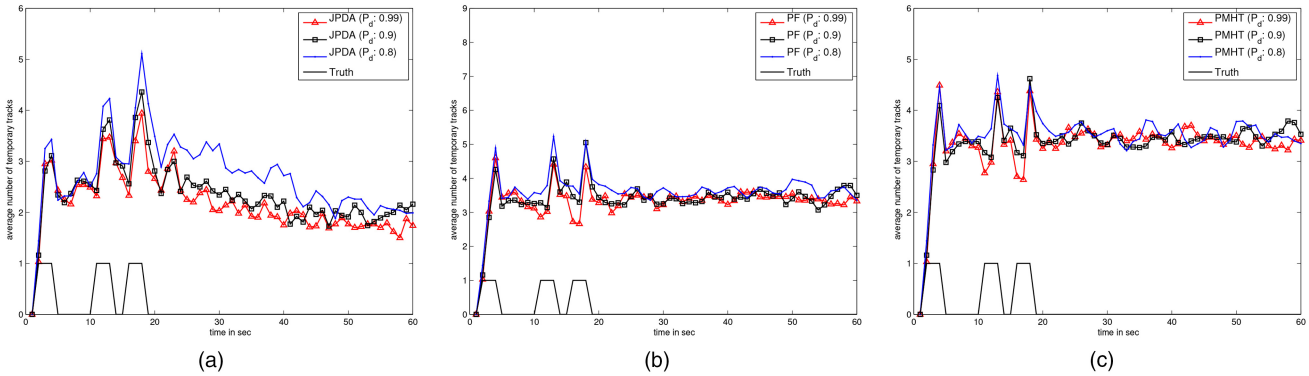


Fig. 7. Average number of temporary tracks with high false alarm density ($\lambda V = 4$): the truth is the number of true tracks considered as temporary tracks following the track confirmation rule. Due to the high false alarm density, the number of temporary tracks apparently increases. (a) JPDAF. (b) Particle Filter. (c) PMHT.

the initiation algorithm, the initial $\mathbf{p}$ and velocity $\mathbf{v}$ are generated from the position $\mathbf{p}_0$ and velocity $\mathbf{v}_0$ of ground truth such that $\mathbf{p} = \mathbf{p}_0 + \omega_1$ and $\mathbf{v} = \mathbf{v}_0 + \omega_2$, where ω_i follows Gaussian distribution $\mathcal{N}(0, \sigma_i^2)$ with $\sigma_1 = \sigma_\gamma$ and $\sigma_2 = 2\sigma_{\dot{\gamma}}$.

For the measurement noise of $\sigma_\gamma = 20$ m and $\sigma_{\dot{\gamma}} = 2$ m/s, the RMSE of the PMHT decreases over time, while the other filters show the opposite tendency (with the exception of the JPDA filter with $P_d = 0.7$), as seen in Figures 8 and 9; presumably this follows from the PMHT's batch-optimization behavior. The performance of the PMHT filter was found to be more sensitive to the illuminator/receiver/target geometry than the other filters. At this level of measurement noise, there was no lost track. It is also noted that the RMSE of the second target is very similar to the third target and is not shown here.

When the range measurement noise is increased to $\sigma_\gamma = 100$ m, the performance of the PMHT for the third target is a little worse than the case having the low measurement noise, but the RMSE of PMHT for the first target with the low detection probability ($P_d = 0.7$)

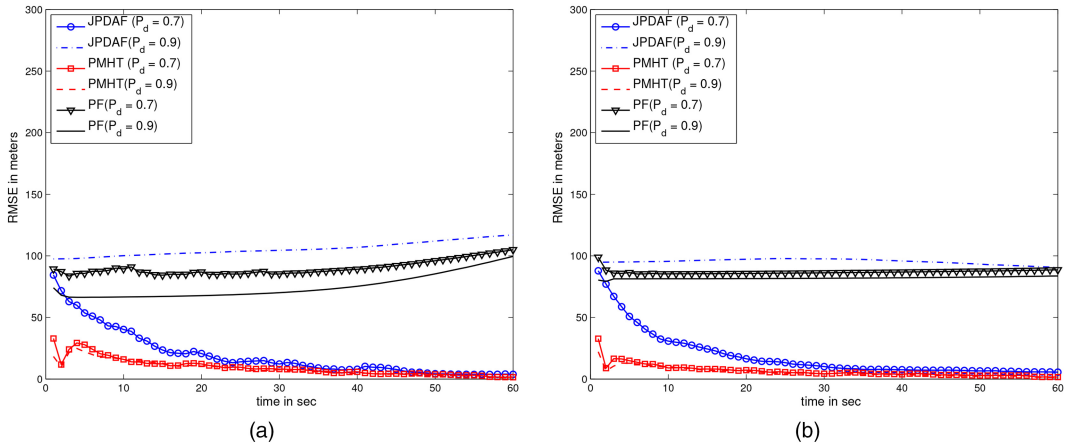


Fig. 8. RMSE for different detection probabilities with $\lambda V = 1$ and low measurement noise ($\sigma_\gamma = 20$ m and $\sigma_{\dot\gamma} = 2$ m/s): the RMSE is less than 150 m, and the RMSE of the PMHT decreases while the other filters increase, but it is sensitive to illuminator/receiver/target geometry. (a) First Target. (b) Third Target.

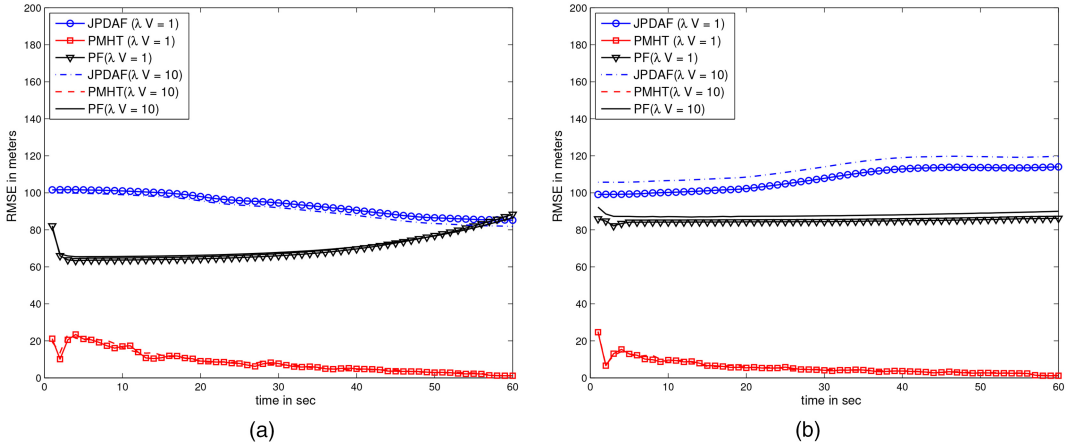


Fig. 9. RMSE for different expected numbers of false alarms (λV) with detection probability $P_d = 0.99$ and low measurement noise ($\sigma_\gamma = 20$ m and $\sigma_{\dot\gamma} = 2$ m/s): the RMSE is less than 180 m and the high false alarm rate has a more profound effect on the PMHT than the low detection probability. (a) First Target. (b) Third Target.

or the high expected number of false alarm ($\lambda V = 10$) is clearly degraded more than other filters, as seen in Figures 10 and 11. The tendency of the RMSE to decrease over time for the PMHT is still present in this case. The rate of track loss in the PMHT filter is less than 2% with detection probability less than 0.8, while there is no lost track in the JPDA filter for every case. The track loss in the particle filter reaches 30% when the detection probability is 0.7. Additionally, the JPDA filter's performance with the low detection probability is better than the high detection probability as seen in Figures 8 and 10, meaning that many measurements degrade the data association rather than help. Further, the performance of the JPDA filter seems unaffected by a higher false alarm density, as seen in Figures 9 and 11. We note that tracker RMSE curves vary significantly with geometry and other factors [5].

C. Discussion

There are separate results for comparisons in terms of track management and in terms of track accuracy. The former are clear and depend little on geometry nor parameter settings: the PMHT is remarkably accurate at determining the number of extant tracks. We have shown typical examples of the latter that strongly imply considerable preferability of the PMHT. But these results depend more on geometry and parameter settings, and it can be difficult to distinguish a track that is being lost from one that is merely bad. Nonetheless, on the basis of these results and others—and also on our observation of computational needs—we confidently contend that the PMHT approach (coupled with the PMHTI for track initiation and simple rules for validation and termination) is the way to go.

It is harder to be confident as to why. We offer the following. All three of these algorithms are approximations. The "super-targets" in the JPDAF's association model are a convenience (plus its "memory" is only one scan); the PMHT uses an "incorrect" assignment prior; and the particle filter both uses a form of the PMHT model and becomes exact only in the limit as the number of particles diverges. It seems that the PMHT's price (modeling) is worth the benefit (multiscan operation).

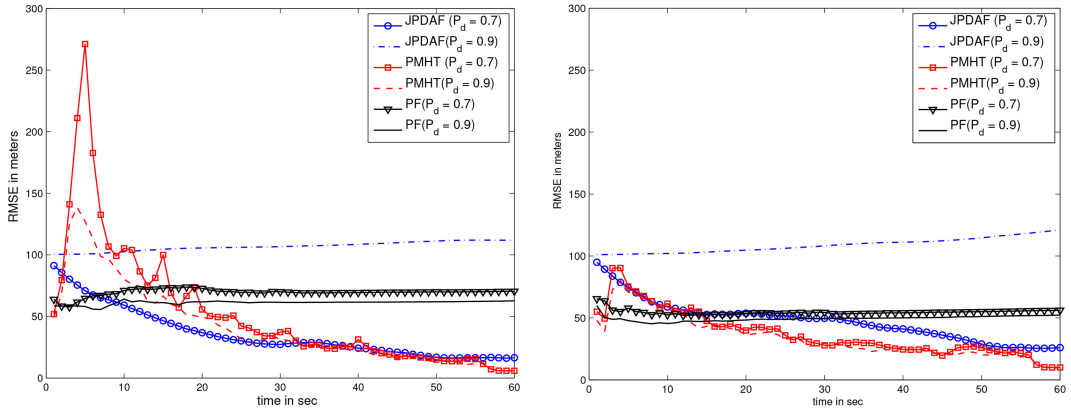


Fig. 10. RMSE for different detection probabilities with high measurement noise (σ_γ = 100 m and $\sigma_{\dot\gamma}$ = 2 m/s): the RMSE is less than 1,000 m. (a) First Target. (b) Third Target.

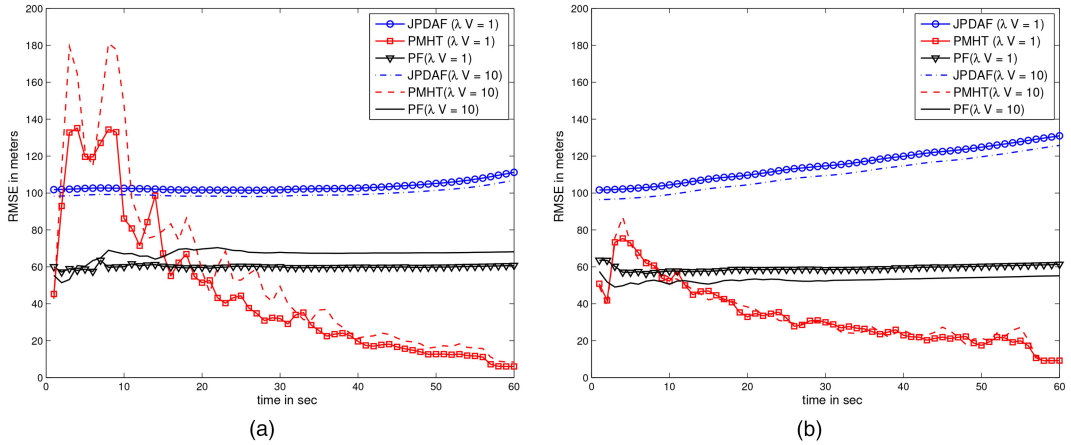


Fig. 11. RMSE for different expected numbers of false alarm (λV) with high measurement noise (σ_γ = 100 m and $\sigma_{\dot\gamma}$ = 2 m/s): the RMSE is less than 1,400 m, and the PMHT filter is more affected by the high false alarm density than other filters. (a) First Target. (b) Third Target.

V. SUMMARY

The design of a passive radar system poaching DAB/DVB signals is challenging, due both to the poor quality of angular information and to the use of indistinguishable signals from multistatic transmitters—there is an additional association ambiguity between measurements and illuminators. A tracking system using the PMHT to deal with the measurement-illuminator-target association was here presented. This (modified) PMHT was compared with a modified JPDA filter and particle filter. When these trackers are combined with the PMHTI method for track-initiation, the PMHT shows excellent performance compared to the other filters. In terms of computational complexity, the PMHT filter is in our experience faster as well.

REFERENCES

[1] Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan.
Estimation with Application to Tracking and Navigation.
YBS publishing, Storrs, CT, 1995.

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

[3] C. Berger, B. Demissie, J. Heckenbach, P. Willett, and S. Zhou.
Signal processing for passive radar using OFDM waveforms.
IEEE Journal on Selected Topics in Signal Processing (special issue on MIMO Radar), 10(5):226–238, 2010.

[4] S. Choi, C. Berger, D. Crouse, P. Willett, and S. Zhou.
Target tracking for multistatic radar with transmitter uncertainty.
In *Proceedings of SPIE:Signal and Data Processing of Small Targets 2009*, volume 7445, 2009.

[5] S. Choi, D. Crouse, P. Willett, and S. Zhou.
Approaches to Cartesian data association passive radar tracking in a DAB/DVB network.
available at the *IEEE Trans. on Aerospace and Electronic Systems*, page doi:10.1117/12.829599.

[6] S. Choi, D. Crouse, P. Willett, and S. Zhou.
Multistatic target tracking for passive radar in a DAB/DVB network: Initiation.
submitted in the *IEEE Trans. on Aerospace and Electronic Systems*.

[7] D. Crouse.
A time-shift model for OFDM radar.
In *The 2012 IEEE Radar Conference*, pages 841–846, Washington, DC, 2012.

[8] D. Crouse, M. Guerriero, and P. Willett.
A critical look at the PMHT.
Journal of Advances in Information Fusion, 4(2):93–116, December 2009.

[9] D. Crouse, M. Guerriero, P. Willett, R. Streit, and D. Dunham.
A look at the PMHT.
In *Proc. of the 12th International conference on Information Fusion*, pages 332–339, Seattle, WA, 2009.

[10] M. Daun and C. Berger.
Track initialization in a multistatic DAB/DVB-T network.
In *Proceeding of the 11th International Conference on Information Fusion*, Cologne (Germany), July 2008.

[11] M. Daun and R. Kaune.
Gaussian mixture initilization in passive tracking applications.
In *Information Fusion, 2010 13th Conference on*, pages 1–8, Rome, Italy, July 2010.

[12] M. Daun and W. Koch.
Multistatic target tracking for non-cooperative illumination by DAB/DVB-T.
In *IEEE Radar Conference*, page DIO 4302238, Rome, Italy, May 2008.

[13] M. Daun, U. Nickel, and W. Koch.
Tracking in multistatic passive radar system using DAB/DVB-T illumination.
Signal Processing, 92(6):1365–1386, June 2012.

[14] E. Giannopoulos, R. Streit, and P. Swaszek.
Probabilistic multi-hypothesis tracking in a multi-sensor, multi-target environment.
In *Data Fusion Symposium, ADFS '96*, pages 184–189, Newport, RI, November 1996.

[15] M. Krieg and D. Gray.
Multi-sensor, probabilistic multi-hypothesis tracking.
In *Data Fusion Symposium, ADFS 96*, pages 153–158, Salisbury, Australia, November 1996.

[16] Streit R. and Luginbuhl T.
Probabilistic multi-hypothesis tracking.
Technical Report 10428, NUWC-NPT, Newport, RI, 1995.

[17] C. Rago, P. Willett, and R. Streit.
Direct data fusion using the PMHT.
In *Proceedings of American Control Conference*, volume 3, pages 1698–1702, Storrs, CT, June 1995.

[18] X. Song, J. Cui, H. Zha, and H. Zhao.
Probabilistic detection-based particle filter for multi-target tracking.
In *Proc. of British Machine Vision Conference*, pages 223–232, 2008.

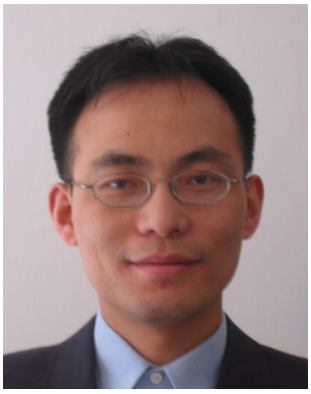
[19] P. Willett, Y. Ruan, and R. Streit.
PMHT: Problems and some solutions.
IEEE Trans. on Aerospace and Electronic Systems, 38(3): 738–754, 2002.

Sora Choi received her Ph.D. degree, in Mathematics, from Seoul National University, Seoul, Korea in 2003. She was a researcher at the Electronics and Telecommunications Research Institute (ETRI), Daejeon, Korea from 2003 to 2004, and was an instructor at Seoul National University between 2000 and 2002, Kyunghee and Konkuk Universities, in Korea, in 2007. She received a Graduate Predoctoral Fellowship in 2011 and Ph.D. in Electrical Engineering at the University of Connecticut, Storrs in 2013. She is currently working in the Department of Electrical Engineering and Computer Science in Syracuse University, Syracuse, NY as a Post Doctorate Research Associate. Her research interests are in sensor fusion, detection theory, target tracking, signal processing, and fault diagnosis.

Peter Willett (F'03) received his B.A.Sc. (Engineering Science) from the University of Toronto in 1982, and his Ph.D. degree from Princeton University in 1986.

He has been a faculty member at the University of Connecticut ever since, and since 1998 has been a Professor. His primary areas of research have been statistical signal processing, detection, machine learning, data fusion and tracking. He has interests in and has published in the areas of change/abnormality detection, optical pattern recognition, communications and industrial/security condition monitoring.

He is editor-in-chief for IEEE Signal Processing Letters and is VP Publications for the IEEE AES Society. He was until recently EIC of the *IEEE Transactions on AES*, and associate editor for *IEEE Transactions on Systems, Man, and Cybernetics*, parts A and B, *IEEE AES Magazine*, and ISIF's *Journal of Advances in Information Fusion*. He was General Co-Chair for the 2006 ISIF/IEEE Fusion Conference in Florence, for 2008 in Cologne and for 2011 in Chicago. He was Program Co-Chair for the 2003 IEEE Conference on Systems, Man, and Cybernetics in Washington DC, and for the 1999 Fusion Conference in Sunnyvale.



Shengli Zhou (IEEE Fellow 2014) received the B.S. degree in 1995 and the M.Sc. degree in 1998, from the University of Science and Technology of China (USTC), Hefei, China, both in electrical engineering and information science. He received his Ph.D. degree in electrical engineering from the University of Minnesota (UMN), Minneapolis, USA, in 2002.

He is now a full professor with the Department of Electrical and Computer Engineering at the University of Connecticut (UCONN), Storrs, USA. He has held a United Technologies Corporation (UTC) Professorship in Engineering Innovation, 2008–2011, and the Charles H. Knapp Associate Professor in Electrical Engineering, 2012–2013. His general research interests lie in the areas of wireless communications and signal processing. His recent focus is on underwater acoustic communications and networking.

Dr. Zhou has served as an Associate Editor for IEEE Transactions on Wireless Communications, 2005–2007, for IEEE Transactions on Signal Processing, 2008–2010, and now is an associate editor for IEEE Journal of Oceanic Engineering. He received the 2007 ONR Young Investigator award and the 2007 Presidential Early Career Award for Scientists and Engineers (PECASE). He is a member of the Connecticut Academy of Science and Engineering (CASE), and has been recently elected to IEEE Fellow.

Practical Data Association for Passive Sensors in 3D

SHUO ZHANG
YAAKOV BAR-SHALOM

This paper considers the passive-sensor data association problem based on multi-dimensional assignment (MDA), a prerequisite for data fusion. The S-D algorithm has been shown to be effective for solving the MDA problem. The bottleneck of the S-D algorithm lies in its cost computation, which consumes about 95%–99% of the CPU times. Since the number of costs in the MDA problem increases exponentially with the number of sensors, the S-D algorithm becomes quite inefficient when a large number of sensors are used. We propose an efficient data association technique, "S_0-D+Seq(2-D)" algorithm, which decomposes the original problem to an S_0-dimensional assignment and several 2-dimensional assignments. The S_0-D+Seq(2-D) algorithm yields a total number of costs which only increases quadratically with the number of sensors. Simulation results show that the S_0-D+Seq(2-D) algorithm achieves a significant reduction in CPU time compared to the S-D algorithm with similar association qualities.

Manuscript received July 25, 2011; revised May 31, 2012 and November 11, 2013; released for publication March 7, 2014.

Refereeing of this contribution was handled by Ramona Georgescu.

Authors' addresses: Shuo Zhang, United Technologies Research Center, East Hartford, CT 06108, E-mail: (zhangs1@utrc.utc.com); Yaakov Bar-Shalom, ECE Department, University of Connecticut, Storrs, CT 06269-2157, U.S.A. E-mail: (shuo.zhang@engr.uconn.edu).

1557-6418/14/$17.00

1. INTRODUCTION

Data association is a crucial task in many surveillance systems and is a prerequisite for data fusion. In general data association solves the correspondence problem in either a "hard" or "soft" manner [2]. A typical step in tracking problems is the measurement-to-track association where it decides which measurement to update which track. There are several state-of-the-art methods in solving the type of association, for example, Joint Probabilistic Data Association (JPDA) and Multiple Hypothesis Tracking (MHT) [2]. In this paper, we consider another type of association called measurement-to-measurement association in a multisensor mutlitarget scenario, where each sensor generates a set of line of sight (LOS) or direction of arrival (DOA), i.e., incomplete position measurements of the targets and the goal is to decide which of the measurements in each sensor correspond to the same target. The measurements are grouped together by an association algorithm and are used to generate a composite (full position) measurement of a common target. In tracking applications, the composite measurements can be used in the subsequent measurement-to-track association to update existing tracks (this is fusion configuration III [2]). This measurement-to-measurement association is considered as "static" where the measurements are assumed to be synchronized, i.e., observed at the same time. The fusion of asynchronous measurements is discussed in [13].

Measurement-to-measurement association becomes especially challenging if the sensors are passive and measure LOS angles only for the targets. Measurements from multiple sensors have to be associated to determine the full positions of the targets. The brute force approach, i.e., enumerating all possible combinations and choosing the most likely one, is computationally prohibitive even for a moderate size problem. For example, the total number of combinations for a scenario of 20 targets and 2 sensors (assuming no missed detections or false alarms) is $20! = 2.4 \times 10^{18}$. A practical approach is to formulate the multisensor data association as a multiple dimensional assignment (MDA) problem [2] and then employ (constrained) optimization techniques to obtain the optimal assignment. When the number of sensors is greater than or equal to three (i.e., $S \geq 3$), the MDA is known to be NP hard. While a number of suboptimal techniques have been proposed, the Lagrangian relaxation based approaches [14], [16] have been shown to be superior to others (e.g., branch and bound, row-column heuristic) for their excellent balance between the accuracy and the efficiency. The relaxation technique in [9] is termed as the S-D (assignment) algorithm. In [18] an extended approach of determining the top m assignments (as opposed to only the best one) has been obtained by using Murty's ranking algorithm [10].

Prior to the optimization step in the S-D algorithm, the first step is to calculate the candidate association

costs. It has been reported [20], [1] that this cost-calculation step consumes 95%–99% of the CPU time. Consequently, when the number of targets is large, a direct use of the S-D algorithm can become quite inefficient. Thus, for the large-scale problem clustering techniques [8] are applied before carrying out the S-D algorithm. By employing the clustering (or partitioning), the original large problem is reduced to a number of smaller subproblems, which can be solved efficiently by the S-D algorithm.

However, even with the clustering, the CPU time of the S-D algorithm increases drastically when a large number of sensors are used. The reason is that the number of costs to be computed increases exponentially with the number of sensors, although this computational burden can be mitigated by employing the gating technique [2], [7] to remove unlikely candidate associations. Note that it is not uncommon to have a large number of passive sensors since some of them have lower costs and are easier for deployment, e.g., infrared or CCD cameras. Aiming at the large-scale data association, that is, when a large number of targets are present and many sensors are used, we propose an efficient data association technique: "S_0-D+Seq(2-D)" algorithm. This algorithm first decomposes the original problem to a (fixed) S_0-dimensional assignment and $S - S_0$ 2-dimensional assignments. Then the former is solved by using the S-D algorithm and the latter is solved by a successive use of the (modified) Auction algorithm [14].

The paper is organized as follows. In Section 2 the MDA problem is formulated for passive sensors in 3D. In Section 3 the iterative least squares (ILS) technique is presented for target position estimation using the LOS measurements. In Section 4 the dihedral-angle based clustering technique is discussed. The proposed S_0-D+Seq(2-D) algorithm is given in Section 5. Simulation results are given in Section 6 based on a large-scale localization problem. Finally, conclusions are given in Section 7.

2. FORMULATION OF THE MDA PROBLEM

Assume that there are S passive sensors in a 3D space, with known positions $\mathbf{p}_s = [x_s, y_s, z_s]'$ $(s = 1, \ldots, S)$. The sensors are assumed to be synchronized.[1] For a given target, each sensor provides its LOS measurement, azimuth angle and elevation angle, namely,

$$z_{i_s} = h(\mathbf{x}, \mathbf{p}_s) + w_s \qquad s = 1, \ldots, S \tag{1}$$

where i_s is the measurement index in sensor s, $\mathbf{x} = [x, y, z]'$ denotes the target's position, w_s is zero-mean white Gaussian measurement noise with covariance R_s and

$$h(\mathbf{x}, \mathbf{p}_s) = \begin{bmatrix} \alpha_s \\ \varepsilon_s \end{bmatrix} = \begin{bmatrix} \tan^{-1}\left(\dfrac{y - y_s}{x - x_s}\right) \\ \tan^{-1}\left(\dfrac{z - z_s}{\sqrt{(x - x_s)^2 + (y - y_s)^2}}\right) \end{bmatrix} \tag{2}$$

Each sensor may receive a number of such measurements from multiple targets, as well as false alarms. An S-tuple of measurements $Z_{i_1 i_2 \ldots i_S}$, consisting of one measurement from each sensor, represents a possible association, that is, the measurements $Z_{i_1 i_2 \ldots i_S}$ are assumed to originate from the same target. Since a target may not be detected by every sensor, a *dummy measurement* is added to each sensor with index 0, to represent the missed detection. If there is only one nondummy measurement in a S-tuple, this nondummy measurement is deemed to be a false alarm.[2] For each S-tuple there is an associated cost $c_{i_1 i_2 \ldots i_S}$, which is given by the negative log-likelihood ratio [2]

$$c_{i_1 i_2 \ldots i_S} = -\ln \frac{\Lambda(Z_{i_1 i_2 \ldots i_S} \mid \mathbf{x})}{\Lambda(Z_{i_1 i_2 \ldots i_S} \mid \emptyset)} \tag{3}$$

The numerator in (3) represents the likelihood that the S-tuple of measurements $Z_{i_1 i_2 \ldots i_S}$ originate from the same target with position $\mathbf{x}$, namely,

$$\Lambda(Z_{i_1 i_2 \ldots i_S} \mid \mathbf{x}) = \prod_{s=1}^{S} [1 - P_{Ds}]^{1 - u(i_s)} [P_{Ds} p(z_{i_s} \mid \mathbf{x})]^{u(i_s)} \tag{4}$$

where P_{Ds} is the detection probability of sensor s, $u(i_s)$ is an indicator function, defined as

$$u(i_s) = \begin{cases} 0 & \text{if } i_s = 0 \\ 1 & \text{otherwise} \end{cases} \tag{5}$$

and $p(z_{i_s} \mid \mathbf{x})$ is given by

$$p(z_{i_s} \mid \mathbf{x}) = |2\pi R_s|^{-1/2} \cdot \exp\left(-\tfrac{1}{2}[z_{i_s} - h(\mathbf{x}, \mathbf{p}_s)]' R_s^{-1} [z_{i_s} - h(\mathbf{x}, \mathbf{p}_s)]\right) \tag{6}$$

The denominator in (3) represents the likelihood that all the measurements in the S-tuple are false alarms, namely,

$$\Lambda(Z_{i_1 i_2 \ldots i_S} \mid \emptyset) = \prod_{s=1}^{S} \lambda^{u(i_s)} \tag{7}$$

where λ denotes the spatial density [2] of the false alarms.

The MDA problem is formulated as follows [9]

$$\min_{\rho_{i_1 i_2 \ldots i_S}} \sum_{i_1=0}^{n_1} \sum_{i_2=0}^{n_2} \cdots \sum_{i_S=0}^{n_S} c_{i_1 i_2 \ldots i_S} \rho_{i_1 i_2 \ldots i_S} \tag{8}$$

[1] This corresponds to Configuration III Fusion, following the classification originated by O. E. Drummond (see [2]).

[2] We make the assumption that a target is detected by at least two sensors, otherwise the target's state is unobservable.

subject to

$$\begin{aligned}
&\sum_{i_2=0}^{n_2}\cdots\sum_{i_S=0}^{n_S}\rho_{i_1 i_2\ldots i_S}=1; \qquad i_1=1,2,\ldots,n_1\\
&\sum_{i_1=0}^{n_1}\cdots\sum_{i_S=0}^{n_S}\rho_{i_1 i_2\ldots i_S}=1; \qquad i_2=1,2,\ldots,n_2\\
&\qquad\qquad\vdots \qquad\qquad \vdots\\
&\sum_{i_1=0}^{n_1}\cdots\sum_{i_{S-1}=0}^{n_{S-1}}\rho_{i_1 i_2\ldots i_S}=1; \qquad i_S=1,2,\ldots,n_S
\end{aligned} \tag{9}$$

where $\rho_{i_1 i_2\ldots i_S}\in\{0,1\}$. Thus, the goal is to find $\{\rho_{i_1 i_2\ldots i_S}\}$, i.e., a partition of the total measurements that minimizes the global cost, subject to the constraints that each measurement is associated with one and only one measurement (including the dummy measurement) in each other sensor. When $S=2$, this MDA problem can be solved exactly by using the modified Auction algorithm [14]. In the general case, i.e., $S>2$, this problem is NP hard and can only be solved suboptimally. The S-D algorithm [9], which is based on the Lagrangian relaxation, has been shown to be an effective technique to solve this general MDA problem.

3. POSITION ESTIMATION VIA ITERATIVE LEAST SQUARES

Since the target position $\mathbf{x}$ in (4) is unknown, it is substituted by its estimate $\hat{\mathbf{x}}$ obtained from the S-tuple of measurements $Z_{i_1 i_2\ldots i_S}$. While there are a number of methods to obtain $\hat{\mathbf{x}}$, the iterative least squares (ILS) technique [3] is preferred since it is easy to implement (no Hessian involved) and provides a (approximate) covariance matrix for its estimate at the same time.

Assume that there are n nondummy measurements in $Z_{i_1 i_2\ldots i_S}$ $(2\le n\le S)$ and we stack them to form an augmented vector $\mathbf{z}$. Then, the ILS estimate in the jth iteration can be written as

$$\begin{aligned}
\hat{\mathbf{x}}^{j+1} &= \hat{\mathbf{x}}^{j}+[(H^j)'R^{-1}H^j]^{-1}\\
&\quad\cdot(H^j)'R^{-1}[\mathbf{z}-\mathbf{h}(\hat{\mathbf{x}},\mathbf{p})]
\end{aligned} \tag{10}$$

where $R=\mathrm{diag}([R_1,\ldots,R_n])$,[3] $\hat{\mathbf{z}}=[\hat{z}_1',\ldots,\hat{z}_n']'$, $\mathbf{h}(\hat{\mathbf{x}},\mathbf{p})=[\mathbf{h}(\hat{\mathbf{x}},\mathbf{p}_1)',\ldots,\mathbf{h}(\hat{\mathbf{x}},\mathbf{p}_n)']'$ and

$$H^j=\left.\frac{\partial\mathbf{h}(\hat{\mathbf{x}},\mathbf{p})}{\partial\mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}^j} \tag{11}$$

is the Jacobian matrix of the stacked measurement vector evaluated at $\hat{\mathbf{x}}^j$. In this case, the Jacobian matrix is

$$H=[H_1'\cdots H_n']' \tag{12}$$

[3]The subscript in R_i is the index of the nondummy measurement and is not the sensor index. It is different from that of the previous section. This holds for other variables.

where

$$H_i=\begin{bmatrix}\dfrac{\partial\alpha_i}{\partial x} & \dfrac{\partial\alpha_i}{\partial y} & \dfrac{\partial\alpha_i}{\partial z}\\ \dfrac{\partial\varepsilon_i}{\partial x} & \dfrac{\partial\varepsilon_i}{\partial y} & \dfrac{\partial\varepsilon_i}{\partial z}\end{bmatrix} \tag{13}$$

$$\frac{\partial\alpha_i}{\partial x}=-\frac{y-y_i}{(x-x_i)^2+(y-y_i)^2} \tag{14}$$

$$\frac{\partial\alpha_i}{\partial y}=\frac{x-x_i}{(x-x_i)^2+(y-y_i)^2} \tag{15}$$

$$\frac{\partial\alpha_i}{\partial z}=0 \tag{16}$$

$$\frac{\partial\varepsilon_i}{\partial x}=-\frac{(x-x_i)(z-z_i)}{\sqrt{(x-x_i)^2+(y-y_i)^2}\|\mathbf{x}-\mathbf{p}_i\|^2} \tag{17}$$

$$\frac{\partial\varepsilon_i}{\partial y}=-\frac{(y-y_i)(z-z_i)}{\sqrt{(x-x_i)^2+(y-y_i)^2}\|\mathbf{x}-\mathbf{p}_i\|^2} \tag{18}$$

$$\frac{\partial\varepsilon_i}{\partial z}=\frac{\sqrt{(x-x_i)^2+(y-y_i)^2}}{\|\mathbf{x}-\mathbf{p}_i\|^2} \tag{19}$$

and $\|\cdot\|$ denotes the Euclidean norm.

To start the ILS recursion an inital estimate $\hat{\mathbf{x}}^0$ is required, which is given by [12]

$$\hat{x}^0=\frac{y_2-y_1+x_1\tan\alpha_1-x_2\tan\alpha_2}{\tan\alpha_1-\tan\alpha_2} \tag{20}$$

$$\hat{y}^0=\frac{\tan\alpha_1(y_2+\tan\alpha_2(x_1-x_2))-y_1\tan\alpha_2}{\tan\alpha_1-\tan\alpha_2} \tag{21}$$

$$\hat{z}^0=z_1+\tan\varepsilon_1\left|\frac{(y_1-y_2)\cos\alpha_2+(x_2-x_1)\sin\alpha_2}{\sin(\alpha_1-\alpha_2)}\right| \tag{22}$$

which has made use of the first two measurements.

4. CLUSTERING

In the S-D algorithm, the most expensive step is computing the association costs, which consumes 95%–99% of the CPU time [20], [1]. From (8) the total number of costs to be calculated is

$$n_c=\prod_{s=1}^{S}(n_s+1) \tag{23}$$

For simplicity, assuming that the number of measurements is n_0 for every list (sensor), then

$$n_c=(n_0+1)^S \tag{24}$$

Consequently, a large n_0 is unfavorable for the efficiency of the S-D algorithm.

The clustering technique is used to reduce a large-size problem to a number of smaller subproblems, which can be solved independently. The clustering algorithm groups measurements based on a distance metric. For the passive sensors in 3D, an effective metric is the so-called dihedral angle [8]. The dihedral angle is defined as the angle between two planes, a target plane and

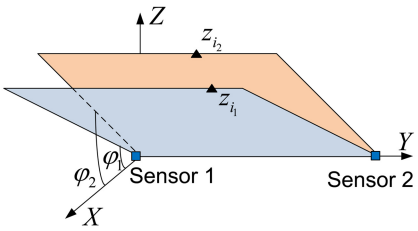


Fig. 1. Clustering using dihedral angles.

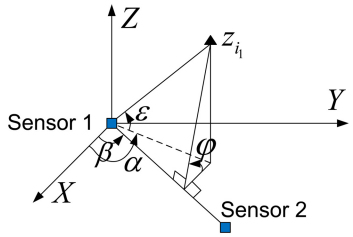


Fig. 2. Computation of the dihedral angle.

TABLE 1
Clustering Using Dihedral Angles

FOR sensor $s = 1 : S - 1$
  FOR sensor $j = s + 1 : S$
    Cluster measurements in sensors s and j using the dihedral angles
  END FOR
  Find the measurements that have not yet been clustered for the use of the next iteration
END FOR

a reference plane (see Fig. 1). The target plane passes through two sensors and one LOS measurement from either of these two sensors, while the reference plane is the XY plane in the 3D Cartesian space (assuming both of the sensors are located on the XY plane). Given two LOS measurements from two different sensors, if these measurements originate from the same target, then the two dihedral angles, one for each LOS measurement, would be close to each other. As a result, clustering the dihedral angles leads to clustering the respective LOS measurements.

The dihedral angle φ for a LOS measurement $[\alpha, \varepsilon]'$ from sensor 1 located at the origin with reference to another sensor 2 located on the same plane as sensor 1, but at an angle β to it (see Fig. 2) is given by

$$\varphi = \tan^{-1}\left(\frac{\tan\varepsilon}{\sin\Delta}\right) \tag{25}$$

where

$$\Delta = \mathrm{mod}(|\alpha - \beta|, \pi) \tag{26}$$

Eq. (26) denotes the angle $|\alpha - \beta|$ with a modulus of π.

The dihedral angles have to be computed in pairwise between each pair of sensors. A summary of the clustering algorithm using the dihedral angles is given in Table 1.

TABLE 2
Numbers of Costs of the S-D algorithm for Different Numbers of Sensors ($n_0 = 7$)

No. of sensors S	No. of costs n_c
4	4,096
7	2,097,152
10	1,073,700,000

The dihedral angle can be also utilized in gating [2] to prune unlikely associations. If a candidate association fails in the gating test, there is no need to compute its cost, i.e., an infinitely large cost is assigned to it. For a given cluster, the calculation of the candidate costs is recursive. Beginning at z_{i_1} in list 1, we take one measurement from each list at a time. If the measurement falls inside the gate defined by the previous measurements in the tuple, this measurement is incorporated in the tuple, which advances to the next list. The cost of the tuple is only evaluated at the last list when a full tuple is achieved. For example, assuming the current list is m and the current association tuple is $Z_{i_1 \ldots i_{m-1}}$, if z_{i_m} passes the gating test then it is added and form $Z_{i_1 \ldots i_m}$, otherwise all the subsequent candidate associations starting with $Z_{i_1 \ldots i_m}$ are discarded. Consequently, the CPU time spent in the cost computation can be saved via this gating process.

REMARK I: While the clustering technique significantly reduces the association complexity, the downside is that some measurements can be grouped incorrectly. Hence the use of clusters poses a design trade-off between complexity and accuracy. Note that the association algorithm to be presented next is not restricted to this clustering technique. It can be integrated with any clustering algorithm for passive sensors in 3D.

5. S_0-D+Seq(2-D) ALGORITHM

For a small number of sensors, the S-D algorithm (along with the clustering technique if the number of targets is large) is able to perform in real time. However, when the number of the sensors increases, the CPU time of the S-D algorithm increases drastically, which is impossible for the S-D algorithm to operate in real time. We can see from (24) the total number of costs increases exponentially with S. For example, assuming n_0 in (24) is 7 (in each cluster we expect a small number of n_0), the numbers of costs for the S-D algorithm are shown in Table 2 for different numbers of sensors.

When $S = 10$, the total number of costs is over one billion[4] for a single cluster.

We propose an efficient data association technique, called S_0-D+Seq(2-D) algorithm, for the case where more than 3 sensors are used. This algorithm consists of two steps, the S_0-D step and the Seq(2-D) step, which are presented next.

[4]The actual number of costs to be computed would be smaller due to gating.

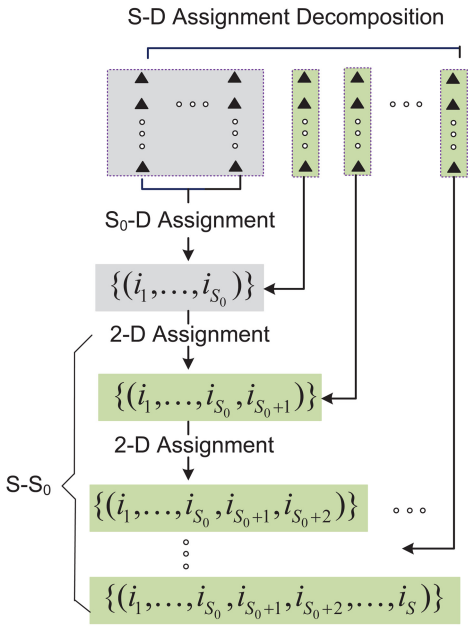


Fig. 3. An illustration of the S_0-D+Seq(2-D) algorithm.

S_0-D step: This steps solves a standard MDA problem with the dimension of S_0 using the S-D algorithm, that is, the data association is performed among S_0 sensors ($S_0 < S$). While the minimum value of S_0 is 2, practically S_0 should be at least 3 to achieve quality associations. This is due to the ghosting problem [2] of the passive sensors. The use of more sensors can mitigate this ghosting effect. A large gate (or no gating) is recommended for the S_0-D step to prevent discarding some less likely (due to noises) but real associations.

Seq(2-D) step: This step solves a series of 2D assignments sequentially using the modified Auction algorithm [14]. The number of the 2D assignments is $S - S_0$. After the S_0-D step, the S_0-tuple association results are available. Then, take a new list from the remaining $S - S_0$ lists and formulate a 2-D assignment between the previous association results and the measurements in this new list. After the 2-D assignment, the length of each association is incremented by one, i.e., becoming a $S_0 + 1$-tuple. Next, take another list from the remaining $S - S_0 - 1$ lists and solve another 2-D assignment, and so on. In the end, after carrying out $S - S_0$ 2-D assignments, each association is in a full tuple, i.e., a S-tuple. The S_0-D+Seq(2-D) algorithm is summarized in Table 3, assuming the S_0-D step chooses the first S_0 lists, i.e., $s = 1,2,\ldots,S_0$. An illustration of the S_0-D+Seq(2-D) algorithm is shown in Fig. 3.

Similarly to (24), the number of costs in the S_0-D step is $(n_0 + 1)^{S_0}$. In the Seq(2-D) step, among the $S - S_0$ 2-D assignments the largest number of costs occur at the last 2-D assignment, which in the worst case is $((S-1)n_0 + 1)(n_0 + 1)$. Consequently, an upper bound of the total number of costs of the S_0-D+Seq(2-D) algorithm is given by

$$n_c' = (n_0 + 1)^{S_0} + (S - S_0)(n_0 + 1)((S-1)n_0 + 1) \quad (27)$$

TABLE 3
S_0-D+Seq(2-D) Algorithm

1) S_0-D step:
   Solve the S_0-D assignment using the S-D algorithm and obtain the S_0-tuple association results $\{(i_1,\ldots,i_{S_0})\}$.
2) Seq(2-D) step:
   FOR $n = S_0 + 1 : S$
      Construct the 2-D assignment between the previous association results $\{(i_1,\ldots,i_{n-1})\}$ and the measurements $\{z_{i_n}\}$ in list n;
      Solve the 2-D assignment using the modified Auction algorithm and obtain the n-tuple results $\{(i_1,\ldots,i_n)\}$.
   END FOR

TABLE 4
Numbers of Costs of the S_0-D+Seq(2-D) algorithm for Different Numbers of Sensors ($n_0 = 7$)

No. of sensors S	No. of costs n_c'
4	688
7	1,888
10	4,096

which increases quadratically with S. In Table 4 the values of this upper bound are shown for different number of sensors.

The association quality of this S_0-D+Seq(2-D) algorithm is evaluated next in the simulation results.

6. SIMULATION RESULTS

We consider a localization problem using the LOS measurements. The numbers of the passive sensors used are 4, 7 and 10. The sensors are located in a circle of radius 5 km centered at (5,5) km in the XY plane, with equal angle separations. The measurement noise standard deviation is 1 mrad for both azimuth and elevation angle. All the sensors are assumed to have the same accuracy, detection probability P_D (= 0.98) and false alarm rate P_F (= 10^{-5}, which corresponds to an average 15 false alarms for each sensor). The number of targets is 300 and their positions are randomly placed in the 3D Cartesian space, where the ranges of the X, Y, Z coordinates are 0–10 km, 1–10 km, 5–10 km, respectively. There is no prior information assumed for the number of targets. The sensor-target geometry is shown in Fig. 4 for the case of 10 sensors.

Given an association tuple, if there is only one nondummy measurement, then it is deemed to represent a false alarm, otherwise it falls into one of the following 3 categories (similar to [1]):

1. Completely correct (CC) association: The measurements in an association tuple have identical origin and there is no dummy measurement associated.
2. Partially correct (PC) association: There are at least 2 measurements with common origin, and the rest may be from different origins or dummy measurements.

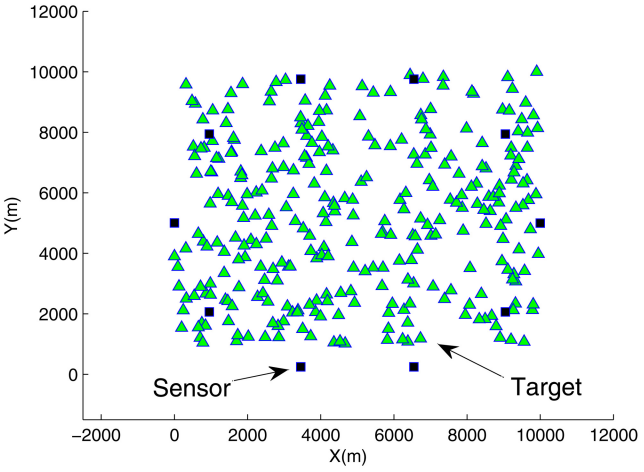


Fig. 4. 300 Targets and 10 sensors (XY projection shown).

3. Completely incorrect (CI) association: In an association tuple, there does not exit a pair of measurements that come from the same origin.

Each CC or PC association corresponds to a detected target (DT). The DT is defined as the origin that appears most in a CC or PC association tuple, and the number of times that the DT appears in a tuple is referred to as the detection index (DI). The detected targets are a subset of the total targets (TT).

Given an association tuple, if the number of the non-dummy measurements is no less than a threshold T_H ($T_H > 1$), then this association is accepted, otherwise it is rejected. To quantify the quality of the accepted associations, we introduce four metrics: fraction of correct associations, fraction of missed targets, fraction of duplicated associations and fraction of purity, which are defined below[5]

- Fraction of correct associations (FCA):
  $$\text{FCA} = \frac{N_{CC} + N_{PC}}{N_{CC} + N_{PC} + N_{CI}}$$
- Fraction of missed targets (FMT):
  $$\text{FMT} = \frac{N_{TT} - N_{DT}}{N_{TT}}$$
- Fraction of duplicated associations (FDA):
  $$\text{FDA} = \frac{N_{CC} + N_{PC} - N_{DT}}{N_{DT}}$$
- Fraction of purity (FP):
  $$\text{FP} = \frac{\overline{DI}}{S}$$

where $\overline{DI}$ denotes the average detection index. Note that only N_{TT} is independent of the threshold T_H.

For the comparison, we consider the method that solves the S dimensional assignment directly, i.e., the S-D algorithm and the proposed sequential method, S_0-D+Seq(2-D), where $S_0 = 3$. The case of $S_0 = 2$ is also evaluated in the simulations. Also, we examine the sequential m-best assignment algorithm, referred to as $m[S_0\text{-D}]+\text{Seq}(m[2\text{-D}])$, where for each S_0-D and 2-D assignment, the top m-best assignments (instead of the only best one) are computed by using the Murty's ranking algorithm [10], [5]. Efficient implementations of the Murty's algorithm can be found in [11], [17], [18]. The $m[S_0\text{-D}]+\text{Seq}(m[2\text{-D}])$ algorithm is performed as follows. For each one of the m solutions obtained from the previous step, the m-best assignment algorithm is carried out, which yields m^2 solutions. Then, m best ones are selected out of these m^2 solutions and stored for the use in the next step. A simplified version, designated as $m[S_0\text{-D}]+\text{Seq}(2\text{-D})$, is also considered in which the m-best assignment is carried out only once for the initial S_0-D assignment to obtain the m best solutions, and for each one of them, the sequential 2-D assignment is used for subsequent associations. All the algorithms are coded in C++ and run on a Intel i7 2.70 GHz laptop. The results are based on 20 Monte Carlo runs.[6]

The dihedral angle based clustering technique from Section 4 is employed. The dihedral angle gating is used when incorporating a new measurement to a association tuple which has been validated before. If the new measurement passes the gate, the resulting new tuple is valid with the tuple length incremented by one, otherwise the new tuple is discarded, i.e., the whole (association) subtree starting with this new tuple is pruned out. The parameters are chosen as: $T_H = 3$, $m = 4$. Although the detection probability is 0.98, a larger value of P_D ($P_D < 1$) is used in (4). This is due to a phenomenon to be called "association splitting," in which a CC or PC association is divided into two or more PC associations, which provide an overall lower cost. A similar phenomenon was observed in [1] for track-to-track associations. This splitting phenomenon will result in incomplete associations.[7] The use of a larger (pseudo) P_D will penalize incomplete associations and prevent the association from splitting.

The CPU times and association qualities for different algorithms are compared in Table 5, where the location RMSE (averaged over all the correct associations) is also provided. The S_0-D+Seq(2-D) algorithm (S_0-D with sequential 2-D), which is, as discussed below, the preferred one, is shown in boldface. The CPU time of the S-D algorithm increases drastically with the number of sensors, S. When $S = 10$, the S-D algorithm requires too much memory that exceeds the computer capacity and no results were obtained. The S_0-D+Seq(2-D) algorithm (S_0-D with sequential 2-D), discussed in Section 5, is advantageous when a large number of sensors are used. When $S = 7$, S_0-D+Seq(2-D) reduces the CPU time of S-D by three orders of magnitude with little difference in association qualities. In terms of either correct

[5] N_X represents the number of X, e.g., N_{DT} denotes the number of detected targets.

[6] This is because a single run of S-D on 7 lists/sensors took half an hour while for 10 lists it became infeasible.

[7] For example, an association of measurements from sensors $\{1,2,3\}$ and one from $\{4,5,6,7\}$ are found "cheaper" (when all of them have the same origin) than associating all of them together.

TABLE 5
Comparison of different algorithms

No. of sensors	Algorithm	FCA	FMT	FDA	FP	RMSE (m)	CPU Time (s)
4	S-D	97.1%	4.8%	3.8%	90.9%	37.6	9.6
4	**S_0-D+Seq(2-D)**	**98.3%**	**6.6%**	**4.1%**	**93.1%**	**39.7**	**2.8**
4	Seq(2-D)	98.6%	10.0%	3.9%	88.7%	42.4	2.0
4	m[S_0-D]+Seq(m[2-D])	97.3%	4.1%	4.2%	92.5%	37.2	8.9
4	m[2-D]+Seq(m[2-D])	98.0%	4.6%	4.1%	92.8%	37.4	9.6
4	m[S_0-D]+Seq(2-D)	97.3%	4.1%	4.2%	92.5%	37.2	7.3
4	m[2-D]+Seq(2-D)	97.6%	4.7%	4.1%	92.7%	37.9	6.9
7	S-D	98.2%	3.6%	6.3%	80.1%	33.9	1316.8
7	**S_0-D+Seq(2-D)**	**97.6%**	**5.2%**	**4.5%**	**86.2%**	**39.8**	**9.9**
7	Seq(2-D)	97.4%	6.6%	4.2%	83.1%	42.7	9.2
7	m[S_0-D]+Seq(m[2-D])	96.8%	2.7%	4.9%	86.7%	35.0	61.1
7	m[2-D]+Seq(m[2-D])	97.7%	3.5%	3.9%	87.1%	35.6	61.8
7	m[S_0-D]+Seq(2-D)	96.8%	3.0%	4.5%	86.9%	35.1	36.4
7	m[2-D]+Seq(2-D)	96.8%	4.0%	4.4%	87.0%	36.4	35.4
10	S-D	—	—	—	—	—	—
10	**S_0-D+Seq(2-D)**	**96.8%**	**4.8%**	**5.3%**	**85.9%**	**40.3**	**20.2**
10	Seq(2-D)	95.8%	5.7%	4.9%	83.3%	44.4	19.5
10	m[S_0-D]+Seq(m[2-D])	96.1%	2.4%	5.5%	86.7%	35.1	169.4
10	m[2-D]+Seq(m[2-D])	97.2%	3.1%	4.1%	87.3%	36.0	169.2
10	m[S_0-D]+Seq(2-D)	96.2%	2.6%	5.2%	86.8%	35.5	75.9
10	m[2-D]+Seq(2-D)	96.3%	3.6%	5.0%	86.9%	37.7	76.4

associations or duplicated associations (FCA and FDA), which one of S-D and S_0-D+Seq(2-D) is better depends on the number of sensors used. The S-D algorithm has fewer missed targets (FMT) and smaller RMSE, while the S_0-D+Seq(2-D) algorithm yields purer associations (FP). With the increase of the number of sensors, both the missed targets and the purities decline for both S-D and S_0-D+Seq(2-D) algorithms. Computationally, Seq(2-D) is the least expensive. S_0-D+Seq(2-D) consumes more CPU time than Seq(2-D) (for large n the time difference is negligible), however, it outperforms the latter in terms of FMT, FP and RMSE. For instance, when $n = 4$ the FMT values imply that S_0-D+Seq(2-D) reduces the missed targets of Seq(2-D) by more than 30%. This is obtained at the cost of extra 40% in CPU time. For large n the improvement is 20% for an extra CPU time of 5–8%.

When $S = 4$, the m[S_0-D]+Seq(m[2-D]) algorithm (m-best S_0-D with sequential m-best 2-D) has the same association qualities as the m[S_0-D]+Seq(2-D) algorithm (m-best S_0-D with sequential 2-D). For $S = 7$ and $S = 10$, m[S_0-D]+Seq(m[2-D]) outperforms m[S_0-D]+Seq(2-D) in terms of FMT and RMSE, but not by a large margin. The corresponding CPU times show that m[S_0-D]+Seq(m[2-D]) is more expensive than m[S_0-D]+Seq(2-D). The CPU time of m[S_0-D]+Seq(2-D) is approximately proportional to m. The m[2-D]+Seq(m[2-D]) algorithm consumes similar CPU time as the m[S_0-D]+Seq(m[2-D]) algorithm, however, the former has degraded performance in terms of FMT and RMSE. The same situation occurs for the m[S_0-D]+Seq(2-D) and m[2-D]+Seq(2-D) algorithms. It is also observed that the (sequential) m-best algorithms slightly outperform the (sequential) algorithms that choose the single best solution in FMT and RMSE, but require more CPU time.

REMARK II: Similarly to the n-scan pruning approach [4] used in the dynamic association, one can also apply a sequential S_0-dimensional assignment to this static association problem, that is, solve the S_0-D assignment on sensors $1,\ldots,S_0$, then make a hard decision on sensor 1 and solve the S_0-D assignment on $2,\ldots,S_0+1$, etc. However, compared to S_0-D+Seq(2-D), at each step the S_0-D assignment (assuming $S_0 > 2$) is more costly than the 2-D assignment for both cost evaluations and optimization. In terms of association performance, from the above results we can see that even the S-D assignment is similar to S_0-D+Seq(2-D), thus the possible improvement of the sequential S_0-D assignment over S_0-D+Seq(2-D) is quite limited.

REMARK III: Although the performance using more sensors appears no better (or worse for some metrics) than using fewer sensors (as the generation of a valid association tuple becomes more demanding), from the robustness point of view the use of more sensors is always beneficial.

7. CONCLUSIONS

This paper presented an efficient data association technique, S_0-D+Seq(2-D) algorithm, for passive sensors in 3D. The passive-sensor data association is a challenging problem, since the line of sight (LOS) measurements from the passive sensors only provide par-

tial knowledge of a target position. The assignment-based methods have been shown to be very effective for data association, where the data association is first formulated as a multiple dimension assignment (MDA) problem and then solved (suboptimally) by using the Lagrangian-relaxation based S-D algorithm. The bottleneck of the S-D algorithm lies in the cost computation, which consumes about 95%–99% of the CPU times. The number of costs to be evaluated in the MDA problem increases exponentially with the number of lists (sensors), which renders the S-D algorithm quite inefficient when a large number of sensors are used. The proposed S_0-D+Seq(2-D) algorithm has a total number of costs increasing quadratically with the number of sensors. As a result, it reduces the number of costs drastically in comparison with the S-D algorithm. For 7 sensors the S_0-D+Seq(2-D) algorithm achieves a CPU time reduction of 3 orders of magnitude compared to the S-D algorithm. The CPU time can be further reduced by introducing parallelization to process different clusters concurrently. The S-D and S_0-D+Seq(2-D) algorithms have similar association qualities. A good choice of S_0 has been shown to be 3. The (sequential) m-best algorithms slightly outperform the (sequential) non m-best algorithms, but are more costly. It has also been shown that the $m[S_0\text{-D}]$+Seq(2-D) algorithm is preferred over the $m[S_0\text{-D}]$+Seq(m[2-D]) algorithm.

REFERENCES

[1] J. Areta, Y. Bar-Shalom, M. Levedahl and K. R. Pattipati
Hierarchical tracking association and fusion for a networked surveillance system.
Journal of Advances in Information Fusion, 1(2):140–156, Dec. 2006.

[2] Y. Bar-Shalom, P. K. Willett and X. Tian
Tracking and Data Fusion.
Storrs, CT: YBS Publishing, 2011.

[3] Y. Bar-Shalom, X. R. Li and T. Kirubarajan
Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software.
New York: Wiley, 2001.

[4] S. S. Blackman
Multiple Hypothesis Tracking For Multiple Target Tracking.
IEEE Aerospace and Electronic Systems Magazine, 19(1):5–18, Jan. 2004.

[5] I. J. Cox and M. Miller
On finding ranked assignments with application to multitarget tracking and motion.
IEEE Transactions on Aerospace and Electronic Systems, 31(1):486–489, Jan. 1995.

[6] I. J. Cox and S. L. Hingorani
An effcient implementation of reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 18(2):138–150, Feb. 1996.

[7] J. B. Collins and J. K. Uhlmann
Efficient gating in data association with multivariate Gaussian states.
IEEE Transactions on Aerospace and Electronic Systems, 28(3):909–916, July 1992.

[8] M. R. Chummun, T. Kirubarajan, K. R. Pattipati and Y. Bar-Shalom
Fast Data Association Using Multidimensional Asignment With Clustering.
IEEE Transactions on Aerospace and Electronic Systems, 37(3), July 2001.

[9] S. Deb, M. Yeddanapudi, K. R. Pattipati and Y. Bar-Shalom
A generalized S-dimensional assignment for multisensor-multitarget state estimation.
IEEE Transactions on Aerospace and Electronic Systems, 33(3):523–538, Apr. 1997.

[10] K. Murty
An algorithm for ranking all the assigments in order of increasing cost.
Operations Research, 16(3):682–687, 1968.

[11] M. L. Miller, H. S. Stone and I. J. Cox
Optimizing Murty's ranked assignment method.
IEEE Transactions on Aerospace and Electronic Systems, 33(3):851–862, July 1997.

[12] R. W. Osborne, III and Y. Bar-Shalom
Statistical efficiency of composite position measurements from passive sensors.
IEEE Transactions on Aerospace and Electronic Systems, 49(4):2799–2806, Oct. 2013.

[13] R. W. Osborne, III, Y. Bar-Shalom and P. Willett
Fusion of Asynchronous Passive Measurements.
17th International conference on information fusion, Salamanca, Spain, July 2014.

[14] K. R. Pattipati, S. Deb, Y. Bar-Shalom and R. B. Washburn
A new relaxation algorithm and passive sensor data association.
IEEE Transactions on Automatic Control, 37(2):198–213, Feb. 1992.

[15] A. B. Poore, N. Rijavec, T. Barker and M. Munger
Data association problems posed as multidimensional assignment problems: algorithm development.
Proc. SPIE Conf. on Signal Processing, Sensor Fusion and Target Recognition, 1955:175–183, 1993.

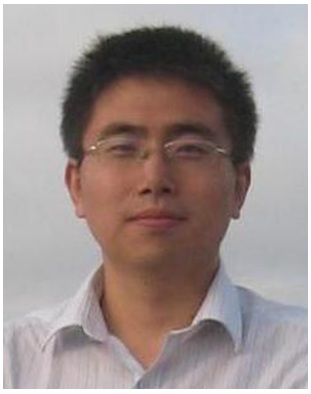
[16] A. B. Poore
Multidimensional Assignment formulation of data association problems arising from multitarget and multisensor tracking.
Computational Optimization and Applications, 3:27–57, 1994.

[17] R. Popp, K. R. Pattipati and Y. Bar-Shalom
An m-Best 2-D assignment algorithm and multilevel parallelization.
Transactions on Aerospace and Electronic Systems, 35(4): 1145–1160, Oct. 1999.

[18] R. L. Popp, K. R. Pattipati and Y. Bar-Shalom
m-best S-D assignment algorithm with application to multitarget tracking.
IEEE Transactions on Aerospace and Electronic Systems, 37(1):22–34, Jan. 2001.

[19] J. K. Uhlmann
Algorithms for multiple-target tracking.
American Scientist, 80:128–141, Mar. 1992.

[20] H. Wang, T. Kirubarajan and Y. Bar-Shalom
Precision large scale air traffic surveillance using IMM/assignment estimators.
IEEE Transactions on Aerospace and Electronic Systems, 35(1):255–266, Jan. 1999.



Shuo Zhang was born in 1982. He received the B.E. degree in 2005 and M.E. degree in 2007 from Harbin Institute of Technology and Ph.D. degree from The University of Connecticut in 2011, all in Electrical and Computer Engineering. His research interests include estimation, perception and convex optimization. Currently He is a Senior Research Scientist at United Technologies Research Center (UTRC). He has published 7 journal papers and more than 10 peer-reviewed conference papers.

Yaakov Bar-Shalom was born on May 11, 1941. He received the B.S. and M.S. degrees from the Technion, Israel Institute of Technology, in 1963 and 1967 and the Ph.D. degree from Princeton University in 1970, all in electrical engineering.

From 1970 to 1976 he was with Systems Control, Inc., Palo Alto, California. Currently he is Board of Trustees Distinguished Professor in the Dept. of Electrical and Computer Engineering and Marianne E. Klewin Professor in Engineering at the University of Connecticut. He is also Director of the ESP (Estimation and Signal Processing) Lab.

His current research interests are in estimation theory, target tracking and data fusion. He has published over 500 papers and book chapters in these areas and in stochastic adaptive control. He coauthored the monograph *Tracking and Data Association* (Academic Press, 1988), the graduate texts *Estimation and Tracking: Principles, Techniques and Software* (Artech House, 1993), *Estimation with Applications to Tracking and Navigation: Algorithms and Software for Information Extraction* (Wiley, 2001), the advanced graduate texts *Multitarget-Multisensor Tracking: Principles and Techniques* (YBS Publishing, 1995), *Tracking and Data Fusion* (YBS Publishing, 2011), and edited the books *Multitarget-Multisensor Tracking: Applications and Advances* (Artech House, Vol. I, 1990; Vol. II, 1992; Vol. III, 2000).

He has been elected Fellow of IEEE for "contributions to the theory of stochastic systems and of multi-target tracking." He has been consulting to numerous companies and government agencies, and originated the series of Multitarget-Multisensor Tracking short courses offered via UCLA Extension, at Government Laboratories, private companies and overseas.

During 1976 and 1977 he served as Associate Editor of the *IEEE Transactions on Automatic Control* and from 1978 to 1981 as Associate Editor of *Automatica*. He was Program Chairman of the 1982 American Control Conference, General Chairman of the 1985 ACC, and Co-Chairman of the 1989 IEEE International Conference on Control and Applications. During 1983–87 he served as Chairman of the Conference Activities Board of the IEEE Control Systems Society and during 1987–89 was a member of the Board of Governors of the IEEE CSS. He was a member of the Board of Directors of the International Society of Information Fusion (1999–2004) and served as General Chairman of FUSION 2000, President of ISIF in 2000 and 2002 and Vice President for Publications in 2004–13.

In 1987 he received the IEEE CSS Distinguished Member Award. Since 1995 he is a Distinguished Lecturer of the IEEE AESS and has given numerous keynote addresses at major national and international conferences. He is co-recipient of the M. Barry Carlton Award for the best paper in the *IEEE Transactions on Aerospace and Electronic Systems* in 1995 and 2000 and recipient of the 1998 University of Connecticut AAUP Excellence Award for Research. In 2002 he received the J. Mignona Data Fusion Award from the DoD JDL Data Fusion Group. He is a member of the Connecticut Academy of Science and Engineering. In 2008 he was awarded the IEEE Dennis J. Picard Medal for Radar Technologies and Applications, and in 2012 the Connecticut Medal of Technology. He has been listed by academic.research.microsoft (top authors in engineering) as #1 among the researchers in Aerospace Engineering based on the citations of his work.

JAIF

Journal of Advances in Information Fusion

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

Volumes 1–8 Index

SUBJECT INDEX

INTERNATIONAL SOCIETY OF INFORMATION FUSION

ISIF Website: http://www.isif.org

2014 BOARD OF DIRECTORS*

2012–2014	2013–2015	2014–2016
Darin T. Dunham	Jean Dezert	Sten F. Andler
Fredrik Gustafsson	Gee-Wah Ng	Murat Efe
Lance M. Kaplan	Anne-Laure Jousselme	Lyudmila Mihaylova

*Board of Directors are elected by the members of ISIF for a three year term.

PAST PRESIDENTS

Wolfgang Koch, 2013	Erik Blasch, 2007	Yaakov Bar-Shalom, 2002
Roy Streit, 2012	Pierre Valin, 2006	Pramod Varshney, 2001
Joachim Biermann, 2011	W. Dale Blair, 2005	Yaakov Bar-Shalom, 2000
Stefano Coraluppi, 2010	Chee Chong, 2004	Jim Llinas, 1999
Elisa Shahbazian, 2009	Xiao-Rong Li, 2003	Jim Llinas, 1998
Darko Musicki, 2008		

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 9 Number 1 June 2014 ISSN 1557-6418