Polynomial-Time Algorithms for the Exact MMOSPA Estimate of a Multi-Object Probability Density Represented by Particles

Marcus Baum, Peter Willett, *Fellow, IEEE*, and Uwe D. Hanebeck

***Abstract*—In multi-object estimation, the traditional minimum mean squared error (MMSE) objective is unsuitable: a simple permutation of object identities can turn a very good estimate into what is apparently a very bad one. Fortunately, a criterion tailored to sets—minimization of the mean optimal sub-pattern assignment (MMOSPA)—has recently evolved. Aside from special cases, exact MMOSPA estimates have seemed difficult to compute. But in this work we present the first exact polynomial-time algorithms for calculating the MMOSPA estimate for probability densities that are represented by particles. The key insight is that the MMOSPA estimate can be found by means of enumerating the cells of a hyperplane arrangement, which is a traditional problem from computational geometry. Although the runtime complexity is still high for the general case, efficient algorithms are obtained for two special cases, i.e., (i) two targets with arbitrary state dimensions and (ii) an arbitrary number of one-dimensional targets.**

***Index Terms*—Target tracking, data association, OSPA distance, Wasserstein distance, MMOSPA estimation.**

I. Introduction

In multi-object estimation cases—these could be "tracking" problems—measurement origin uncertainty necessarily gives rise to uncertainty in target labeling. For example, consider two targets in one-dimensional space as in Fig. 1 that move according to the trajectory depicted in Fig. 2(a). Noisy measurements of the target locations are available. However, it is unknown which measurement comes from which target. A consequence of the unknown measurement origin is that the likelihood function is symmetric in the target states as shown Fig. 3(a). A standard approach to tracking the targets is to employ a particle filter [2], [7], [20], [21], [25], [29], i.e., the posterior density is approximated with particles. When the targets are still separated in the beginning, the posterior joint density for the target locations might look like the one in Fig. 3(b): here each particle comprises the two (scalar) coordinates of *both* targets, and since the targets are initially well separated this figure has all particles with target 1 on the right and target 2 on the left. After the targets have been close,

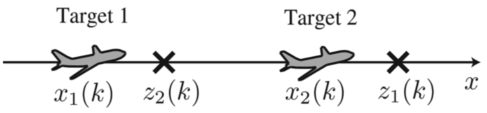
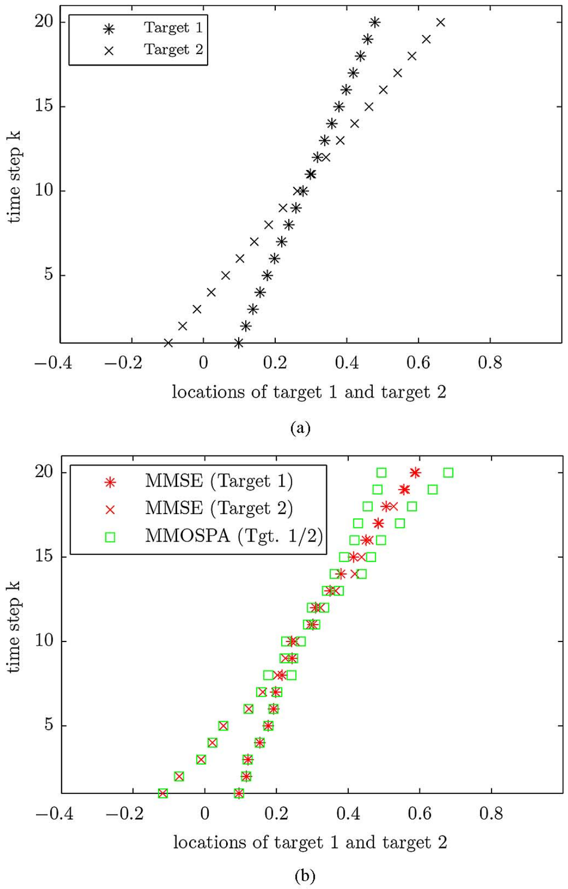


Fig. 1. Two objects at time k with one-dimensional locations $x_1(k)$ and $x_2(k)$ from which two measurements $z_1(k)$ and $z_2(k)$ are received (with unknown measurement origin).



(a)

(b)

Fig. 2. Motivating example: Tracking two one-dimensional targets with a particle filter. (a) Two targets in one-dimensional space with a straight-line motion. (b) Point estimates based on a particle filter for an example run. The MMSE estimate suffers from coalescence once targets crossed.

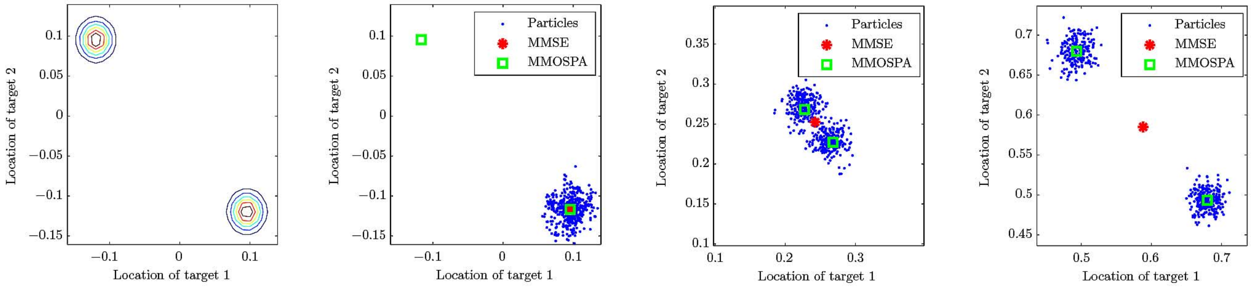
Manuscript received June 11, 2014; revised November 03, 2014; accepted January 19, 2015. Date of publication February 12, 2015; date of current version April 09, 2015. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Amir Asif. This research was partially supported by ONR under contracts N00014-10-10412 and N00014-13-1-0231. M. Baum was partially supported by the Karlsruhe House of Young Scientists (KHYS).

M. Baum and U. D. Hanebeck are with the Intelligent Sensor-Actuator-Systems Laboratory (ISAS), Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology (KIT), Karlsruhe 76131, Germany (e-mail: marcus.baum@kit.edu; uwe.hanebeck@ieee.org).

P. Willett is with the Department of Electrical and Computer Engineering, University of Connecticut, Storrs, CT 06269-2157 USA (e-mail: willett@engr.uconn.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2015.2403292



(a) (b) (c) (d)

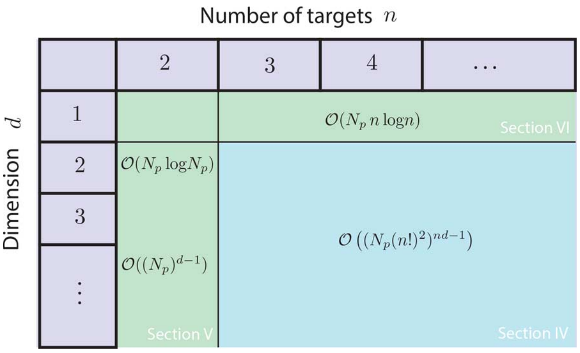
Fig. 3. Motivating example: Likelihood $p(z_1(k), z_2(k)|x_1(k), x_2(k))$ and posterior joint density of the locations of the two targets $p(x_1(k), x_2(k)|\mathcal{Z}(k))$ for different time steps k, where $\mathcal{Z}(k)$ denotes all measurements up to time k. Note, as MMOSPA ignores labels, each permutation of the MMOSPA estimate is also a MMOSPA estimate. For example, in (b) are two (equivalent) MMOSPA estimates $[-0.08, 0.08]^T$ and $[0.08, -0.08]^T$. (a) Likelihood function: $k = 1$. (b) Joint density: $k = 1$. (c) Joint density: $k = 10$. (d) Joint density: $k = 20$.

the true posterior becomes multimodal due to the lost labeling information, see Fig. 3(c). Apparently this is irrecoverable (see Fig. 3(d)), since now approximately half the particles have target 1 on the left and target 2 on the right; and half the other way. The resulting MMSE estimate is in the middle—and quite useless.

This misbehavior leads to the coalescence of the estimated tracks [8], [9] as illustrated in Fig. 2(b). As a remedy, one could use a maximum a-posteriori (MAP) estimator. However, the MAP estimator must choose between modes (and, hence, ignore information about the other mode), which can lead to undesired jittering [10]. An alternative is to eschew any pretense of estimating the (labeled) locations of target 1 and target 2, and simply to ask: "Where are there objects?" To seek an MMSE estimate makes no sense in this case, since MSE requires labels. However, we consider the path to the MMSE estimate: As will be soon discussed, there has recently arisen a *label-free metric*, the optimal sub-pattern assignment (OSPA) metric [26], upon which the criterion mean OSPA (MOSPA) and optimal minimum MOSPA (MMOSPA) approach can be built [23]. The corresponding MMOSPA estimates for the joint densities in the particle filter example are illustrated in Fig. 3. It can be seen that the MMOSPA estimate always captures the locations of the targets precisely. This is also reflected in the estimated trajectory in Fig. 2(b), which does not suffer from coalescence.

During the last years, MMOSPA estimation found plenty of applications. Besides track extraction and display [10], approximations in multi-target tracking algorithms can be performed with an eye to OSPA, as in the *Set JPDAF* [28], the *Set MHT* [10], and several improved multi-object particle filters [2], [7], [20], [21]. A further recent application is direction-of-arrival estimation of signals from multiple targets impinging on an antenna array [4], [12], [13].

The MMSE estimate has an explicit expression, the mean of the posterior; MMOSPA estimation is more complex. Prior work proposes an iterative algorithm [23] to get an approximate solution. An explicit expression was derived for the two-object Gaussian case; and a decent approximation is available for Gaussian mixtures. In [10], [12], [14], a greedy suboptimal algorithm for probability densities represented with particles was introduced. Furthermore, the problem was formulated as an S-D assignment problem in [12] and continuous quadratic semi-assignment formulations were given in [12], [15]. A closed-form approximation of the MOSPA cost function for Gaussian mixtures was presented in [16].



Number of targets n

Dimension d

	2	3	4	…
1		$\mathcal{O}(N_p\, n \log n)$ Section VI		
2	$\mathcal{O}(N_p \log N_p)$	$\mathcal{O}\left((N_p(n!)^2)^{nd-1}\right)$		
3				
⋮	$\mathcal{O}((N_p)^{d-1})$ Section V	Section IV		

Fig. 4. Overview: Runtime complexity of the exact algorithms depending on the number of particles N_p, number of objects n, and state dimension d. The efficient cases are obtained for $n = 2$ and $d = 1$.

For a fixed number of targets, it has been shown in [5] that the MMOSPA estimate for probability densities represented with particles is equivalent to the Wasserstein barycenter [1], [17], [24] for point clouds. Hence, for example, the sliced Wasserstein distance proposed in [24] could be used to get an approximate solution. Explicit solutions are available only for the scalar case [12], [24] and for two particles with arbitrary target number [6].

A. *Contribution*

Suppose one has a particle representation of a multi-object posterior density at a specific time, e.g., Fig. 3(c), and could evaluate MOSPA of some estimate: what should one do with them to find the *best* estimate, the MMOSPA? In this article, we explore *exact* solutions for cases in which probability densities are represented by particles, see the overview in Fig. 4. We present the first exact algorithm for an arbitrary number of targets with a runtime complexity that is polynomial in the number of particles. The runtime complexity for many targets might still be to high for practical applications. However, we further consider two important special cases, i.e., the two-target case and the one-dimensional case, in which further optimizations are possible in order to obtain efficient real-time capable algorithms.

This article extends and revises the conference paper [3], where exact algorithms for the one-dimensional case and the two-target case were discussed.

B. Overview

In the next section, we give an introduction to MMOSPA estimation including the final optimization problem for probability distributions represented by particles. Section III summarizes basic results from computational geometry about hyperplane arrangements, which are essential for deriving the exact algorithms. Subsequently, in Section IV, we present the exact MMOSPA algorithm for the general case, i.e., arbitrary target number and target state dimension. Section V considers the two-target case in which further runtime optimizations are possible. Section VI treats the special case of an arbitrary number of one-dimensional objects and shows that—in this case—the MMOSPA estimate can be calculated efficiently by means of order statistics. We give an extensive evaluation for the exact algorithm for two targets in Section VII. This article is concluded in Section VIII.

II. Problem Description

We consider multi-object estimation problems in which the state of multiple objects is modeled as a random vector

$$\underline{\boldsymbol{x}}(k) = [\underline{\boldsymbol{x}}_1^T(k), \ldots, \underline{\boldsymbol{x}}_n^T(k)]^T, \tag{1}$$

where $\underline{\boldsymbol{x}}_i(k) \in \mathrm{R}^d$ denotes the state vector of target i (with $1 \leq i \leq n$) and k is the time index.

Recursive Bayesian tracking algorithms compute a probability density function for $\underline{\boldsymbol{x}}(k)$ conditioned on all available measurements $\mathcal{Z}(k)$

$$p(\underline{x}(k) \,|\, \mathcal{Z}(k)). \tag{2}$$

Due to the nonlinearity of the data association problem, usually only an approximation of (2) is available. In this work, we focus on particle filters [25], [29], i.e., the probability density (2) is represented with N_p particles

$$\boxed{p(\underline{x}(k) \,|\, \mathcal{Z}(k)) = \sum_{i=1}^{N_p} w_i(k) \cdot \delta(\underline{x}(k) - \underline{x}^{(i)}(k)),} \tag{3}$$

where $\underline{x}^{(i)}(k)$ denotes the i-th particle with weight $w_i(k)$ and $\delta(\,\cdot\,)$ is the Dirac delta function.

Remark 1: In the remainder of this work, we focus on a single point in time k, so that we omit the time index, i.e., instead of $x_1(k)$ we will write x_1.

In general, a point estimate minimizes a specific risk function such as the mean square error (MSE), which yields the minimum mean square error (MMSE) estimate

$$\underline{\hat{x}}^{\mathrm{MMSE}} := \operatorname*{arg\,min}_{\underline{\hat{x}} \in \mathrm{R}^{nd}} \mathrm{E}\left\{ \|\underline{\hat{x}} - \underline{\boldsymbol{x}}\|^2 \,|\, \mathcal{Z} \right\}.$$

It is well-known that the MMSE estimate is given by the conditional mean $\underline{\hat{x}}^{\mathrm{MMSE}} = \mathrm{E}\{\underline{\boldsymbol{x}} \,|\, \mathcal{Z}\}$.

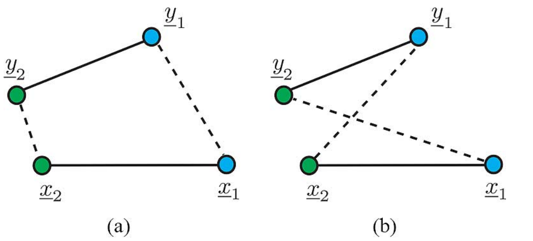


Fig. 5. OSPA distance for two two-dimensional targets $\underline{x} = [\underline{x}_1^T, \underline{x}_2^T]^T$ and $\underline{y} = [\underline{y}_1^T, \underline{y}_2^T]^T$: There are two possible permutations π_1 and π_2, i.e., $\mathrm{OSPA}(\underline{x}, \underline{y})^2 = \frac{1}{2}\min\{\|\underline{x} - P_{\pi_1}(\underline{y})\|^2, \|\underline{x} - P_{\pi_2}(\underline{y})\|^2\}$. (a) Permutation π_1. (b) Permutation π_2.

As motivated in the introduction, the MMSE estimate may be a poor choice for multi-object estimation since its implicit *constraint* of object labeling. As an alternative, we abandon the target labels by using a different risk function than the MSE, namely the mean optimal sub-pattern assignment (MOSPA) [23], [28]. This risk function is based on the OSPA metric [26], which is a widely-used measure for evaluating multi-target tracking algorithms. In this work, we only need the special case of a known number of targets, see Fig. 5.

Definition 1 (OSPA): The *optimal sub-pattern assignment (OSPA)* [26] distance between two vectors $\underline{x} = [\underline{x}_1^T, \ldots, \underline{x}_n^T]^T$ and $\underline{y} = [\underline{y}_1^T, \ldots, \underline{y}_n^T]^T$, which consist of n target state vectors of dimension d, is defined as

$$\mathrm{OSPA}(\underline{x}, \underline{y})^2 = \frac{1}{n} \min_{\pi \in \Pi_n} \|\underline{x} - P_\pi(\underline{y})\|^2, \tag{4}$$

where Π_n denotes all permutations of the set $\{1, \ldots, n\}$, and $P_\pi(\underline{y})$ permutes the single target states in $\underline{y}$ according to π, i.e., $P_\pi(\underline{y}) = [\underline{y}_{\pi(1)}^T, \cdots, \underline{y}_{\pi(n)}^T]^T$.

For example, for two one-dimensional targets ($n = 2, d = 1$), we have the squared error $\mathrm{SE}([10\,20]^T, [19\,11]^T) = 162$, $\mathrm{OSPA}([10\,20]^T, [19\,11]^T)^2 = 2$.

Remark 2: We will use the notation $P_{\underline{x}}(\underline{y}) := P_{\pi_{\mathrm{opt}}}(\underline{y})$ for the optimal reordered single target states in $\underline{y}$ with respect to $\underline{x}$ (in order to emphasize that π_{opt} depends on $\underline{x}$). Then, we can equivalently write for (4)

$$\mathrm{OSPA}(\underline{x}, \underline{y})^2 = \frac{1}{n}\|\underline{x} - P_{\underline{x}}(\underline{y})\|^2. \tag{5}$$

Based on the OSPA metric, we introduce the MMOSPA estimator [23].

Definition 2 (MMOSPA): The *minimum mean optimal sub-pattern assignment (MMOSPA)* [23] estimate of $\underline{\boldsymbol{x}} = [\underline{\boldsymbol{x}}_1^T, \ldots, \underline{\boldsymbol{x}}_n^T]^T$ based on the posterior probability density $p(\underline{x}|\mathcal{Z})$ is defined as the point estimate that minimizes the mean OSPA (MOSPA) distance, i.e.,

$$\underline{\hat{x}}^{\mathrm{MMOSPA}} := \operatorname*{arg\,min}_{\underline{\hat{x}} \in \mathrm{R}^{nd}} \mathrm{E}\{\mathrm{OSPA}(\underline{\hat{x}}, \underline{\boldsymbol{x}})^2 \,|\, \mathcal{Z}\}_{p(\underline{x}|\mathcal{Z})}. \tag{6}$$

There are $n!$ MMOSPA estimates, because each permutation of $\underline{\hat{x}}^{\mathrm{MMOSPA}}$ is also a MMOSPA estimate. Due to the properties of the expectation, (6) can be written as [23]

$$\underline{\hat{x}}^{\mathrm{MMOSPA}} = \mathrm{E}\{P_{\underline{\hat{x}}^{\mathrm{MMOSPA}}}(\underline{\boldsymbol{x}}) \,|\, \mathcal{Z}\}. \tag{7}$$

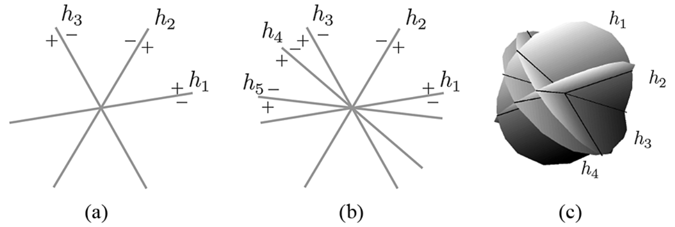


Fig. 6. Central arrangements of hyperplanes in 2D with three hyperplanes (a) and five hyperplanes (b). In 2D, the number of cells increases linearly with the number of hyperplanes. A 3D hyperplane arrangement is shown in (c). In 3D, the number of cells grows quadratically with the number of hyperplanes.

Note that $\hat{\underline{x}}^{\text{MMOSPA}}$ is unknown in the right hand-side of (7), which is why MMOSPA is complicated to compute.

For densities represented with particles as (3), the MMOSPA is given by [11]

$$\hat{\underline{x}}^{\text{MMOSPA}} = \operatorname*{arg\,min}_{\hat{\underline{x}} \in \mathrm{R}^{\text{nd}}} \sum_{i=1}^{N_p} w_i \cdot \text{OSPA}(\hat{\underline{x}}, \underline{x}^{(i)})^2 \tag{8}$$

$$= \operatorname*{arg\,min}_{\hat{\underline{x}} \in \mathrm{R}^{\text{nd}}} \sum_{i=1}^{N_p} w_i \cdot \frac{1}{n} \| \hat{\underline{x}} - P_{\pi^{(i)}_{\text{opt}}}(\underline{x}^{(i)}) \|^2 , \tag{9}$$

where $\pi^{(i)}_{\text{opt}}$ is the optimal permutation of the i-th particle with respect to $\hat{\underline{x}}$. Using the notation of (5), we obtain

$$\boxed{\hat{\underline{x}}^{\text{MMOSPA}} = \operatorname*{arg\,min}_{\hat{\underline{x}} \in \mathrm{R}^{\text{nd}}} \sum_{i=1}^{N_p} w_i \cdot \tfrac{1}{n} \| \hat{\underline{x}} - P_{\hat{\underline{x}}}(\underline{x}^{(i)}) \|^2 ,} \tag{10}$$

which is the final formulation of the optimization problem that we will solve in this article. According to (7), the MMOSPA estimate can also be written as

$$\boxed{\hat{\underline{x}}^{\text{MMOSPA}} = \sum_{i=1}^{N_p} w_i \cdot P_{\hat{\underline{x}}^{\text{MMOSPA}}} \left(\underline{x}^{(i)} \right) .} \tag{11}$$

III. Brief Introduction to Hyperplane Arrangements

This section summarizes some basic results from computational geometry about hyperplane arrangements that are relevant for this work (see for example [18], [19], [22], [27]).

A hyperplane arrangement in d-dimensional space is given by a set

$$\mathcal{A} = \{ h_i \,|\, i = 1, \ldots, N \}$$

consisting of N hyperplanes

$$h_i := \{ a \in \mathrm{R}^d \,|\, \underline{b}_i^T a = c_i \} , \tag{12}$$

where $\underline{b}_i \in \mathrm{R}^d$ and $c_i \in \mathrm{R}$. If all hyperplanes pass through the origin, i.e., $c_i = 0$ for all $i = 1, \ldots, N$, the arrangement is called *central* (see Fig. 6). A d-dimensional *central* hyperplane arrangement is equivalent to a $d-1$-dimensional *non-central* one, see [27].

Each hyperplane partitions R^d into three sets, i.e., the hyperplane h_i, and the two half-spaces

$$h_i^+ = \left\{ \underline{a} \in \mathrm{R}^d \,|\, \underline{b}_i^T \underline{a} - c_i \geq 0 \right\} , \text{ and}$$
$$h_i^- = \left\{ \underline{a} \in \mathrm{R}^d \,|\, \underline{b}_i^T \underline{a} - c_i \leq 0 \right\} .$$

An arrangement is composed of d-dimensional regions we shall call *cells*. Each point $\underline{a} \in \mathrm{R}^d$ can be labeled with a sign vector $\gamma(\underline{a}) \in \{-, 0, +\}^N$, whose i-th component is

$$(\gamma(\underline{a}))_i := \begin{cases} - & \text{if } \underline{a} \in h_i^- \\ 0 & \text{if } \underline{a} \in h_i \\ + & \text{if } \underline{a} \in h_i^+ \end{cases} . \tag{13}$$

Hence, each cell of an arrangement can be uniquely represented by a sign vector $\underline{s} \in \{-, +\}^{N_p}$. A d-dimensional hyperplane arrangement with N hyperplanes has at most [22]

$$\sum_{i=0}^{d} \binom{N}{d-i} \in \mathcal{O}(N^d) \tag{14}$$

cells. This property will become essential later as it says that the number of cells grows polynomially in the number of hyperplanes.

Algorithms for enumerating the cells of a d-dimensional hyperplane arrangement are described in [18], [19], [22], [27]. There are algorithms that are optimal in the sense that they enumerate each cell exactly once with a worst-case runtime $\mathcal{O}(N^d)$. For higher dimensions, the optimal algorithms are rather technically complex. However, an extremely simple general algorithm that is not time-optimal [22] is shown in Algorithm 1. First, all vertices in the arrangement are calculated. Second, the cells adjacent to the vertices are enumerated. The obtained runtime is $\mathcal{O}(N^{d+1})$. In the following, we focus on non-degenerate hyperplane arrangements, i.e., any d hyperplanes intersect in a unique vertex and any $d+1$ hyperplanes have no common points [22], as degenerate hyperplanes are zero-probability events.

Algorithm 1: Cell enumeration of non-central hyperplane arrangement $\mathcal{A} = \{ h_i \,|\, i = 1, \ldots, N \}$ according to [22]

```
$\mathcal{S} = \{\}$
for all vertices $\underline{v}$ in $\bigcap_i \underline{h}_i$ do
    $\underline{s} := \gamma(\underline{v})$
    $S := \mathcal{S} \cup \text{adjacentCells}(\underline{s})$
end for
return $\mathcal{S}$
```

IV. General Polynomial Time Algorithm

In this section, we derive a polynomial-time algorithm for computing the exact MMOSPA estimate (10).

Remark 3: A naive but inherently intractable approach is to check all $n!^{N_p}$ possible permutations of the particles in (11) and find the permutations that yield the least MOSPA distance.

We pursue the approach to directly optimize over all $\hat{\underline{x}} \in \mathrm{R}^{\text{nd}}$ in (10). We will show that there is only a polynomial number of possible vectors $\hat{\underline{x}}$ to check. Specifically, we show that all potential vectors can be found by enumerating the cells of a hyperplane arrangement, which is a well-known problem in geometry and combinatorics [27]. In general, the resulting worst-case complexity of the derived exact algorithm for d-dimensional target states will be

$$\mathcal{O}((N_p \cdot (n!)^2)^{\mathrm{nd}-1}), \tag{15}$$

where N_p is the number of particles and n the number of targets. Hence, the complexity is polynomial in the number of particles N_p, while it is exponential in the number of targets n.

The first step is to consider the OSPA distance in (4): In order to calculate the OSPA distance between two vectors $\underline{x}$ and $\underline{y}$, one has to consider all permutations $P_\pi(\underline{x})$ (with $\pi \in \Pi_n$) of the single target states in $\underline{x}$. The following theorem gives a simplified condition to check which of two permutations π_1 and π_2 is "better". The most important property of this condition is that it is linear in $\underline{y}$, which will later allow us to construct a hyperplane arrangement.

Theorem 1: For the two vectors $\underline{x} = [\underline{x}_1^T, \ldots, \underline{x}_n^T]^T$ and $\underline{y} = [\underline{y}_1^T, \ldots, \underline{y}_n^T]^T$, which consist of n targets with dimension d, and two permutations $\pi_1, \pi_2 \in \Pi_n$, the following holds:

$$\frac{1}{n}\|\underline{y} - P_{\pi_1}(\underline{x})\|^2 \leq \frac{1}{n}\|\underline{y} - P_{\pi_2}(\underline{x})\|^2 \tag{16}$$

$$\Updownarrow$$

$$0 \leq \langle \underline{y}, P_{\pi_2}(\underline{x}) - P_{\pi_1}(\underline{x})\rangle \tag{17}$$

Proof:

$$\frac{1}{n}\|\underline{y} - P_{\pi_1}(\underline{x})\|^2 \leq \frac{1}{n}\|\underline{y} - P_{\pi_2}(\underline{x})\|^2 \tag{18}$$

$$\Updownarrow$$

$$\langle \underline{y} - P_{\pi_1}(\underline{x}), \underline{y} - P_{\pi_1}(\underline{x})\rangle \leq \langle \underline{y} - P_{\pi_2}(\underline{x}), \underline{y} - P_{\pi_2}(\underline{x})\rangle \tag{19}$$

$$\Updownarrow$$

$$-2\langle \underline{y}, P_{\pi_1}(\underline{x})\rangle \leq -2\langle \underline{y}, P_{\pi_2}(\underline{x})\rangle \tag{20}$$

$$\Updownarrow$$

$$0 \leq \langle \underline{y}, P_{\pi_1}(\underline{x}) - P_{\pi_2}(\underline{x})\rangle, \tag{21}$$

where (20) follows from (19) because for all $\pi \in \Pi_n$ we have $\langle P_\pi(\underline{x}), P_\pi(\underline{x})\rangle = \langle \underline{x}, \underline{x}\rangle$. □

Next, we will use the above theorem to construct a hyperplane arrangement for a particle distribution (3), which is the main insight that leads to the polynomial-time algorithm.

Theorem 2: Suppose we are given the random vector $\underline{\boldsymbol{x}} = [\underline{\boldsymbol{x}}_1^T, \ldots, \underline{\boldsymbol{x}}_n^T]^T \in \mathrm{R}^{nd}$, which consists of n-dimensional single target states and a corresponding posterior density (3). Then there is a set of nd-dimensional vectors

$$A = \{\underline{a}_1, \ldots, \underline{a}_{N_a}\} \subset \mathrm{R}^{nd} \tag{22}$$

that contains a specific $\hat{\underline{a}} \in A$ with

$$\hat{\underline{x}}^{\mathrm{MMOSPA}} = \sum_{i=1}^{N_p} w_i P_{\hat{\underline{a}}}(\underline{x}^{(i)}). \tag{23}$$

The cardinality N_a of A is bounded by a polynomial depending on the number of particles N_p, i.e.,

$$N_a \in \mathcal{O}((N_p \cdot (n!)^2)^{nd-1}).$$

Proof:
Step 1: Hyperplane arrangement for a single particle.
Based on Theorem 1, define for each particle $\underline{x}^{(i)}$ an nd-dimensional central hyperplane arrangement

$$\mathcal{A}^{(i)} = \bigcup_{1 \leq r < s \leq n!} h_i^{r,s} \tag{24}$$

that consists of $\frac{n!(n!-1)}{2}$ hyperplanes

$$h_i^{r,s} := \{\underline{a} \in \mathrm{R}^{nd} \mid (\underline{b}_i^{r,s})^T\, \underline{a} = 0\}, \tag{25}$$

with

$$\underline{b}_i^{r,s} := P_{\pi_r}\left(\underline{x}^{(i)}\right) - P_{\pi_s}\left(\underline{x}^{(i)}\right) \tag{26}$$

for all $1 \leq r < s \leq n!$. Here, π_r and π_s enumerate all possible permutations in Π_n.

Intuitively, for a given vector $\underline{a} \in \mathrm{R}^{nd}$, the hyperplane $h_i^{r,s}$ allows to check if π_r or π_s is a "better" permutation for calculating $\mathrm{OSPA}(\underline{a}, \underline{x}^{(i)})$.

Due to Theorem 1 the following holds: If $\underline{a}_1$ and $\underline{a}_2$ are in the same cell, i.e., $\gamma(\underline{a}_1) = \gamma(\underline{a}_2)$, then $P_{\underline{a}_1}(\underline{x}^{(i)}) = P_{\underline{a}_2}(\underline{x}^{(i)})$.
Step 2: Hyperplane arrangement for all particles.
Next, we create, based on all particles, an nd-dimensional central hyperplane arrangement

$$\mathcal{A} = \bigcup_{i=1,\ldots,N_p} \mathcal{A}^{(i)} \tag{27}$$

that consists of $N_p \frac{n!(n!-1)}{2}$ hyperplanes. Each cell uniquely specifies the permutations of all particles, and hence,

$$\sum_{i=1}^{N_p} w_i P_{\underline{a}_1}\left(\underline{x}^{(i)}\right) = \sum_{i=1}^{N_p} w_i P_{\underline{a}_2}\left(\underline{x}^{(i)}\right). \tag{28}$$

If we enumerate the N_a cells of $\mathcal{A}$ with $\mathcal{C}_1, \ldots, \mathcal{C}_{N_a}$, we can pick $\underline{a}_i \in \mathcal{C}_i$. Furthermore, we know from (14) that

$$N_a \leq \left(N_p \frac{n!(n!-1)}{2}\right)^{nd-1}, \tag{29}$$

which is polynomial in the number of particles N_p. □

The cells specified by the vectors (22) can also be specified by sign vectors such as (13). In this manner, let $\underline{s} = \gamma(\underline{a})$ be the sign vector specifying the cell that contains $\underline{a}$ in the hyperplane arrangement $\mathcal{A}$. Then, we denote with $P_{\underline{s}}(\underline{x}^{(i)})$ the permutation that corresponds to $P_{\underline{a}}(\underline{x}^{(i)})$.

Theorem 2 directly leads to an algorithm for calculating the exact MMOSPA estimate: Enumerate the cells in the hyperplane arrangement (27). For each cell, determine the corresponding permutations in (11) and test if it minimizes the corresponding MOSPA distance in (10). This algorithm is summarized in Algorithm 2. As there are optimal algorithms for cell enumeration, the runtime complexity is given by (15). Note that the MOSPA distance for a cell in line 3 does not have to be computed from scratch if the cells are enumerated in a proper order, i.e., previous cell was a neighbor.

Remark 4: The sign vectors of a central arrangement of hyperplanes are symmetric, i.e., $\underline{s} \in \gamma(\mathrm{R}^d) \Rightarrow -\underline{s} \in \gamma(\mathrm{R}^d)$. Hence, we actually would only have to enumerate half of the

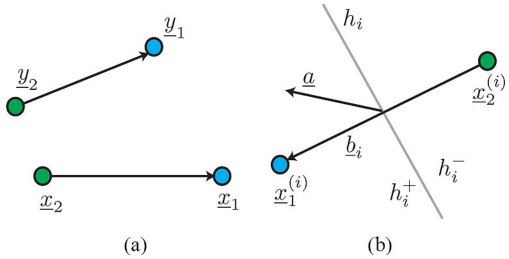


Fig. 7. Optimization over directions. (a) Illustration of Theorem 3. Both vectors $\underline{x}_1 - \underline{x}_2$ and $\underline{y}_1 - \underline{y}_2$ point to the same direction, i.e., $\langle \underline{x}_1 - \underline{x}_2, \underline{y}_1 - \underline{y}_2 \rangle > 0$. Hence, no switching is performed. (b) Illustration of Theorem 4. For all $\underline{a} \in h_i^+$ no switching of $\underline{x}_1^{(i)}$ and $\underline{x}_2^{(i)}$ is performed and for all $\underline{a} \in h_i^-$, the target states are switched.

cells, because the resulting MMOSPA estimates are the same for $\underline{s}$ and $-\underline{s}$. For the sake of simplicity, this property is not exploited here.

Algorithm 2: MMOSPA estimate for n d-dimensional targets with joint density $p(\underline{x}|\mathcal{Z}) = \sum_{i=1}^{N_p} w_i \cdot \delta(\underline{x} - \underline{x}^{(i)})$

```
1: Create the hyperplane arrangement $A$ as defined by (27).
2: $S$ = set of sign vectors specifying the cells $A$.
3: $\underline{s}_{min} = \text{argmin}_{\underline{s} \in S}\{\text{MOSPA}(p(\underline{x}|\mathcal{Z}), \hat{\underline{x}}^{\underline{s}})\}$,
     where $\hat{\underline{x}}^{\underline{s}} := \sum_{i=1}^{N_P} w_i P_{\pi^s}(\underline{x}^{(i)})$
4: return $\hat{\underline{x}}^{\underline{s}_{\min}}$
```

V. Special Case: Two d-Dimensional Targets

In case of two targets it is possible to further improve the runtime complexity of the exact algorithm. The key insight for this purpose is that in the two-target case, it is sufficient to optimize over all direction vectors in order to find the MMOSPA estimate. By this means, the problem is reduced to enumerating the cells of a d-dimensional hyperplane arrangement, instead of a $2d$-dimensional hyperplane arrangement (as it would be in Theorem 2 for $n = 2$).

For $d > 2$, the worst-case complexity is $\mathcal{O}(N_p^{d-1})$, where N_p is the number of particles. For $d = 2$, we give a concrete (simple) algorithm with complexity $\mathcal{O}(N_p \log N_p)$.

The following theorem states that the optimal permutation for computing the OSPA distance between two state vectors for two targets is fully determined by two direction vectors (see Fig. 7(a)). This theorem is actually a two-target version of Theorem 1.

Theorem 3: For two vectors $\underline{x} = [\underline{x}_1^T, \underline{x}_2^T]$ and $\underline{y} = [\underline{y}_1^T, \underline{y}_2^T]^T$, which consist of two targets with dimension d, and the two permutations π_1 (do not switch targets) and π_2 (switch targets) $\in \Pi_2$, the following holds:

$$\frac{1}{2}||\underline{y} - P_{\pi_1}(\underline{x})||^2 \leq \frac{1}{2}||\underline{y} - P_{\pi_2}(\underline{x})||^2 \tag{30}$$

$$\Updownarrow$$

$$0 \leq \langle \underline{x}_2 - \underline{x}_1, \underline{y}_2 - \underline{y}_1 \rangle. \tag{31}$$

Intuitively, the target states in $\underline{x}$ are switched if $\underline{x}$ and $\underline{y}$ point in opposite directions.

Proof: Due to the properties of the scalar product we obtain

$$\begin{aligned} ||\underline{x}_1 - \underline{y}_1||^2 + ||\underline{x}_2 - \underline{y}_2||^2 &\leq ||\underline{x}_1 - \underline{y}_2||^2 + ||\underline{x}_2 - \underline{y}_1||^2 \\ &\Updownarrow \\ -2\langle \underline{x}_1, \underline{y}_1 \rangle - 2\langle \underline{x}_2, \underline{y}_2 \rangle &\leq -2\langle \underline{x}_1, \underline{y}_2 \rangle - 2\langle \underline{x}_2, \underline{y}_1 \rangle \\ &\Updownarrow \\ 0 &\leq \langle \underline{x}_1 - \underline{x}_2, \underline{y}_1 - \underline{y}_2 \rangle. \end{aligned}$$

Definition 3: For $\underline{x} = [\underline{x}_1^T, \underline{x}_2^T] \in \mathrm{R}^{2d}$ and vector $\underline{a} \in \mathrm{R}^d$, we define the operation

$$\text{sort}_{\underline{a}}(\underline{x}) = \begin{cases} [\underline{x}_1^T, \underline{x}_2^T]^T & \text{if } \underline{a}^T(\underline{x}_1 - \underline{x}_2) \geq 0 \\ [\underline{x}_2^T, \underline{x}_1^T]^T & \text{otherwise} \end{cases},$$

which reorders the vector $\underline{x}$ according to the direction of $\underline{a}$.

It follows from the above theorem that the MMOSPA estimate is uniquely specified by a direction vector.

Lemma 1: For two targets, the MMOSPA estimate $\hat{\underline{x}}^{\text{MMOSPA}}$ can be written as

$$\hat{\underline{x}}^{\text{MMOSPA}} = \mathrm{E}\{\text{sort}_{\underline{a}^{\text{MMOSPA}}}(\underline{\boldsymbol{x}}) \,|\, \mathcal{Z}\},$$

where $\underline{a}^{\text{MMOSPA}} := \frac{\hat{\underline{x}}_2^{\text{MMOSPA}} - \hat{\underline{x}}_1^{\text{MMOSPA}}}{||\hat{\underline{x}}_2^{\text{MMOSPA}} - \hat{\underline{x}}_1^{\text{MMOSPA}}||}$.

Proof: The MMOSPA estimate is given by

$$\begin{aligned} \hat{\underline{x}}^{\text{MMOSPA}} &= \mathrm{E}\{P_{\hat{\underline{x}}^{\text{MMOSPA}}}(\underline{\boldsymbol{x}})|\mathcal{Z}\} \\ &= \mathrm{E}\{\text{sort}_{\underline{a}^{\text{MMOSPA}}}(\underline{\boldsymbol{x}}) \,|\, \mathcal{Z}\}. \end{aligned}$$

□

A consequence of this theorem is that finding the MMOSPA estimate for two targets in d-dimensional space is a $(d-1)$-dimensional optimization problem over all possible direction vectors $\underline{a} \in S^d$, where the set of all direction vectors is denoted $S^d := \{\underline{x} \in \mathrm{R}^d \,|\, ||\underline{x}|| = 1\}$. In other words: the direction vector that yields the minimum MOSPA distance has to be found. The next theorem shows that it is only necessary to try a polynomial number of directions in case of particles. This theorem is essentially a special case of Theorem 2. The only difference is that the hyperplanes for the general case of n-targets in Theorem 2 are nd-dimensional. Here, for the two target case ($n = 2$), it is sufficient to consider d-dimensional hyperplanes.

Theorem 4: Suppose we are given the random vector $\underline{\boldsymbol{x}} = [\underline{\boldsymbol{x}}_1^T, \underline{\boldsymbol{x}}_2^T]^T$, which consists of two d-dimensional single target states and a corresponding posterior density (3). Then there is a set of d-dimensional vectors

$$A = \{\underline{a}_1, \ldots, \underline{a}_{N_a}\} \subset \mathrm{R}^d \tag{32}$$

that contains a specific $\hat{\underline{a}} \in A$ with

$$\hat{\underline{x}}^{\text{MMOSPA}} = \sum_{i=1}^{N_p} w_i \text{sort}_{\hat{\underline{a}}}(\underline{x}^{(i)})$$

and the number of direction vectors N_a can be bounded by a polynomial depending on the number of particles N_p, i.e.,

$$N_a \in \mathcal{O}((N_p)^{d-1}).$$

Proof: Each particle $\underline{x}^{(i)} \in \mathrm{R}^{2d}$ defines a hyperplane in R^d

$$h_i := \left\{ a \in \mathrm{R}^d \,|\, \underline{b}_i^T a = 0 \right\}, \tag{33}$$

where $\underline{b}_i := \underline{x}_1^{(i)} - \underline{x}_2^{(i)}$ is the vector that connects the two target states, see Fig. 7(b) for a visualization. In this manner, we obtain a central hyperplane arrangement

$$\mathcal{A} = \{h_i \,|\, i = 1, \ldots, N_p\}. \tag{34}$$

The key observation is that $\mathrm{sort}_a(\underline{x}^{(i)})$ is the same for all direction vectors $\underline{a}$ in a particular half-space h_i^+ and h_i^- according to Theorem 3. Hence, from $\underline{a}_1, \underline{a}_2 \in S^d$ with $\gamma(\underline{a}_1) = \gamma(\underline{a}_2)$ it follows that $\mathrm{sort}_{\underline{a}_1}(\underline{x}^{(i)}) = \mathrm{sort}_{\underline{a}_2}(\underline{x}^{(i)})$ for all $i \in \{1, \ldots, N_p\}$, which yields $\mathrm{E}\{\mathrm{sort}_{\underline{a}_1}(\underline{\boldsymbol{x}}) \,|\, \mathcal{Z}\} = \mathrm{E}\{\mathrm{sort}_{\underline{a}_2}(\underline{\boldsymbol{x}}) \,|\, \mathcal{Z}\}$.

As a consequence, we can enumerate the cells $\mathcal{C}_1, \ldots, \mathcal{C}_{N_a}$ and pick $\underline{a}_i \in \mathcal{C}_i$. Furthermore, the number of cells of a central arrangement of N_p hyperplanes is known to be bounded by (see Section III) $N_a \in \mathcal{O}((N_p)^{d-1})$. □

Again, it is not necessary to explicitly compute the set A, because a sign vector $\underline{s} \in \{-,+\}^{N_p}$ is sufficient to determine the permutation of the particles. For the two-target case, we can explicitly write

$$P_{\underline{s}}\left(\underline{x}^{(i)}\right) := \begin{cases} [\underline{x}_1, \underline{x}_2]^T & \text{if } (\underline{s})_i = + \\ [\underline{x}_2, \underline{x}_1]^T & \text{if } (\underline{s})_i = - \end{cases}$$

for all $i \in \{1, \ldots, N_p\}$.

The final algorithm for the two target case is given in Algorithm 3, where it essentially only differs from Algorithm 2 in the constructed hyperplane arrangement. In the following, we further discuss the 2D and 3D case as these are the most important special cases for two targets, e.g., for displaying the tracks.

Algorithm 3: MMOSPA estimate for two d-dimensional targets with joint density $p(\underline{\boldsymbol{x}} \,|\, \mathcal{Z}) = \sum_{i=1}^{N_p} w_i \cdot \delta(\underline{\boldsymbol{x}} - \underline{\boldsymbol{x}}^{(i)})$

```
Create the hyperplane arrangement $A$ as defined by (34).
$S =$ set of sign vectors specifying the cells $A$.
$\underline{s}_{\min} = \mathrm{argmin}_{\underline{s} \in S}\{\mathrm{MOSPA}(p(\underline{x}|\mathcal{Z}), \hat{\underline{x}}^{\underline{s}})\}$,
    where $\hat{\underline{x}}^{\underline{s}} := \sum_{i=1}^{N_P} w_i P_{\underline{s}}(\underline{x}^{(i)})$
**return** $\hat{\underline{x}}^{\underline{s}_{\min}}$
```

A. Two-Dimensional States

For 2D target states, a very simple algorithm for enumerating the cells can be found, because hyperplanes (through the origin) in 2D can be ordered according to their orientation. This observation is used in Algorithm 4 to compute the MMOSPA estimate for two 2D target states.

The complexity for sorting the hyperplanes is $\mathcal{O}(N_p \, \log N_p)$. For each cell, the corresponding MOSPA distance has to be computed (Line 8 in Algorithm 4). Hence, the overall complexity of Algorithm 4 is $\mathcal{O}(N_p^2)$ with a naive implementation of line 8. Due to the ordering of the hyperplanes, the MOSPA distance does not have to be computed from scratch for each $\underline{s} \in S$ and line 8 can be implemented in linear time. As a consequence, the overall complexity becomes $\mathcal{O}(N_p \, \log N_p)$.

Algorithm 4: Two two-dimensional targets: MMOSPA estimate for $p(\underline{\boldsymbol{x}} \,|\, \mathcal{Z}) = \sum_{i=1}^{N_p} w_i \cdot \delta(\underline{\boldsymbol{x}} - \underline{\boldsymbol{x}}^{(i)})$

```
1: **Assumption:** $\underline{x}^{(i)}$ are sorted such that $\beta_1 \leq \ldots \leq \beta_{N_p}$,
   where $\beta_i \in (0, \pi)$ is the angle between the $x$-axis and
   hyperplane $\underline{h}_i$ (as defined in (33)).
2: $\underline{s} = \gamma([1, 0]^T)$          // Sign vector of first cell
3: $\mathcal{S} = \{\underline{s}\}$
4: **for** $i = 1, \ldots, N_p - 1$ **do**
5:     $(\underline{s})_i := -(\underline{s})_i$          // Sign vector of next cell
6:     $\mathcal{S} = \mathcal{S} \cup \{\underline{s}\}$
7: **end for**
8: $\underline{s}_{\min} = \mathrm{argmin}_{\underline{s} \in S}\{\mathrm{MOSPA}(p(\underline{x}|\mathcal{Z}), \hat{\underline{x}}^{\underline{s}})\}$,
   where $\hat{\underline{x}}^{\underline{s}} := \sum_{i=1}^{N_P} w_i \cdot P_{\underline{s}}(\underline{x}^{(i)})$
9: **return** $\hat{\underline{x}}^{\underline{s}_{\min}}$
```

B. Three-Dimensional States

In order to compute the MMOSPA estimate for two three-dimensional target states, it is necessary to enumerate the cells of a 3D central hyperplane arrangement. A 3D central arrangement of hyperplanes can be considered as 2D non-central hyperplane arrangement by means of intersection with the 3D unit sphere S^3 (see Fig. 6(c)). Hence, we can use the optimal topological sweep algorithm as described in [18].

VI. Special Case: One-Dimensional States

The second relevant special case is that of one-dimensional targets: there is an intuitive connection between MMOSPA estimation and order statistics here. The reason is that the OSPA distance for one-dimensional target states can be computed by sorting the state vectors.

Theorem 5: The OSPA metric for two vectors $\underline{x} = [x_1, \ldots, x_n]^T$ and $\underline{y} = [y_1, \ldots, y_n]^T$ of n one-dimensional target states is given by

$$\mathrm{OSPA}(\underline{x}, \underline{y})^2 = \frac{1}{n} ||\mathrm{sort}(\underline{x}) - \mathrm{sort}(\underline{y})||^2, \tag{35}$$

where the operator $\mathrm{sort}(\cdot)$ sorts the elements of a vector in ascending order.

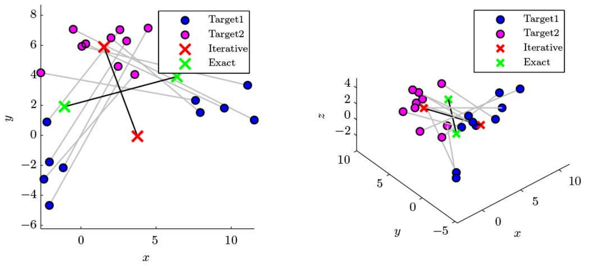
Proof: See for example [3], [24], [30]. □

Based on the above theorem, we can relate order statistics to MMOSPA estimation.

Theorem 6: The MMOSPA estimate for a random $\underline{\boldsymbol{x}} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]^T$ consisting of n one-dimensional targets is given by the mean of the posterior order statistics of $\underline{\boldsymbol{x}}$

$$\hat{x}^{\mathrm{MMOSPA}} = \mathrm{E}\{\mathrm{sort}(\underline{\boldsymbol{x}}) \,|\, \mathcal{Y}\}. \tag{36}$$

Proof: W.l.o.g., let the MMOSPA estimate $\hat{x}^{\mathrm{MMOSPA}}$ be sorted (there are $n!$ MMOSPA estimates and one of them is sorted). Then $\mathrm{sort}(\underline{\boldsymbol{x}})$ coincides with $P_{\underline{x}^{\mathrm{MMOSPA}}}(\underline{\boldsymbol{x}})$ according to Theorem 5. □



(a) (b)

$N_p = 10$

	Naive	Iterat.	Exact
Time	0.7814	0.0012	0.0007
Error	0.0000	14.7667	0.0000
Optimal	1.0000	0.5000	1.0000

$N_p = 100$

	Naive	Iterat.	Exact
Time	-	0.0085	0.0022
Error	-	19.4023	0.0000
Optimal	-	0.2500	1.0000

$N_p = 500$

	Naive	Iterat.	Exact
Time	-	0.0481	0.0089
Error	-	19.8045	0.0000
Optimal	-	0.2500	1.0000

(c)

$N_p = 10$

	Naive	Iterat.	Exact
Time	0.7763	0.0014	0.0135
Error	0.0000	8.1393	0.0000
Optimal	1.0000	0.5500	1.0000

$N_p = 100$

	Naive	Iterat.	Exact
Time	-	0.0106	1.0996
Error	-	16.4810	0.0000
Optimal	-	0.3000	1.0000

$N_p = 500$

	Naive	Iterat.	Exact
Time	-	0.0519	35.6430
Error	-	19.2366	0.0000
Optimal	-	0.2500	1.0000

(d)

Fig. 8. Evaluation results for the naive exact algorithm, novel exact algorithm, and iterative algorithm (for 20 different randomly drawn particles distributions with $N_p = 10$, $N_p = 100$, and $N_p = 500$ particles). The *Time* row gives the average runtime (in seconds, non-optimized MATLAB code, Intel Core i5 processor). The *Error* column is the average squared OSPA distance between the exact and calculated estimate. The *Optimal* row shows the relative number of trials where the optimal result has been obtained. Due to its exponential runtime complexity the naive algorithm could only be used for $N_p = 10$ particles. (a) Example particles for the evaluation with two 2D targets ($N_p = 10$). (b) Example particles for the evaluation with two 3D targets ($N_p = 10$). (c) Detailed results for two 2D targets. (d) Detailed results for two 3D targets.

In case of a particle representation, (36) can be computed by sorting the particles (see Algorithm 5), which has a linear runtime $\mathcal{O}(N_p \, n \, \log n)$ in the number of particles N_p. This algorithm is known in literature in the context of computer vision, see for example [24].

```
Algorithm 5: $n$ targets in 1D:
MMOSPA estimate for $p(\underline{x}|\mathcal{Y}) = \sum_{i=1}^{N_p} w_i \cdot \delta(\underline{x} - \underline{x}^{(i)})$
    return $\sum_{i=1}^{N_P} w_i \cdot \mathrm{sort}(\underline{x}^{(i)})$
```

VII. Evaluation

In this section, a numerical evaluation of the proposed exact algorithm is presented for the two target case[1]. The exact MMOSPA estimate is compared with the naive exact algorithm (see Remark 3), and the iterative algorithm [23]. The iterative algorithm is initialized with the mean of the density as suggested in [23].

a) Two-Dimensional States: We consider two targets with two dimensional states ($n = 2, d = 2$) for which a Gaussian mixture density for the posterior (2) is given. The Gaussian mixture consists of three equally weighted Gaussians with means $[-2, -3, 4, 6]^T$, $[4, 1, 3, 8]^T$, and $[10, 2, 0, 5]^T$; and identical covariance matrices $2\mathbf{I}_2$, where $\mathbf{I}_2$ is the two dimensional identity matrix. We obtain three different particle approximations (3) by sampling from the Gaussian mixture with $N_p = 10$, $N_p = 100$, and $N_p = 500$ particles. Fig. 8(a) shows an example and Fig. 8(c) presents detailed results averaged over 20 trials. The algorithms are compared with regards to runtime and approximation quality. It can clearly be seen that the iterative algorithm sometimes gets stuck in local minima. Also, the runtime of the exact algorithm is significantly lower than the iterative algorithm. Hence, the exact algorithm outperforms the iterative algorithm in all disciplines. The naive exact algorithm is intractable except for $N_p = 10$ particles.

b) Three-Dimensional States: We performed the same experiment with three-dimensional state vectors ($n = 2, d = 3$) and a Gaussian mixture consisting of three equally weighted Gaussian with means $[-2, -3, 1, 4, 6, 0]^T$, $[4, 1, 2, 3, 8, 1]^T$, and $[10, 2, 1, 0, 5, 1]^T$; and identical covariance matrices $2\mathbf{I}_3$, where $\mathbf{I}_3$ is the three-dimensional identity matrix. Again, the naive exact algorithm is intractable and the iterative algorithm tends to suffer from local minima, see Fig. 8(b) and (d). However, the iterative algorithm is now faster than the exact algorithm.

VIII. Conclusion

MMOSPA estimation has become a fundamental concept in multi-object estimation, where the traditional MMSE estimator is meaningless. In this article, we developed algorithms for the case that the probability densities are represented by particles. A main (and probably surprising) result is that it is possible to calculate exact MMOSPA estimates with a polynomial runtime in the number of particles (and hence *not exponential*). These algorithms provide an efficient (real-time capable) exact solution for the two target case and for the one-dimensional case. For the general case, the algorithms might be useful for an off-line performance assessment of approximate algorithms.

References

[1] M. Agueh and G. Carlier, "Barycenters in the Wasserstein space," *SIAM J. Math. Anal.*, vol. 43, no. 2, pp. 904–924, 2011.

[2] E. H. Aoki, Y. Boers, L. Svensson, P. Mandal, and A. Bagchi, "A Bayesian look at the optimal track labelling problem," in *Proc. 9th IET Data Fusion Target Track. Conf. (DFTT 2012): Algorithms Appl.*, 2012, pp. 1–6.

[3] M. Baum, P. Willett, and U. D. Hanebeck, "Calculating some exact MMOSPA estimates for particle distributions," in *Proc. 15th Int. Conf. Inf. Fusion (Fusion)*, Singapore, Jul. 2012, pp. 847–853.

[4] M. Baum, P. Willett, and U. D. Hanebeck, "MMOSPA-based direction-of-arrival estimation for planar antenna arrays," in *Proc. 8th IEEE Sensor Array Multichannel Signal Process. Workshop (SAM)*, A Coruña, Spain, Jun. 2014, pp. 209–212.

[5] M. Baum, P. K. Willett, and U. D. Hanebeck, "On Wasserstein barycenters and MMOSPA estimation," *IEEE Signal Process. Lett.*, vol. 22, no. 10, pp. 1511–1515, Oct. 2015.

[6] A. Bishop, "Information fusion via the Wasserstein barycenter in the space of probability measures: Direct fusion of empirical measures and Gaussian fusion with unknown correlation," in *Proc. 17th Int. Conf. Inf. Fusion*, Salamanca, Spain, 2014, pp. 1–7.

[7] H. A. P. Blom and E. Bloem, "Optimal decomposed particle filtering of two closely spaced Gaussian targets," in *Proc. 50th IEEE Conf. Decision Control-Eur. Control Conf. (CDC-ECC)*, 2011, pp. 7895–7901.

[8] H. Blom, E. Bloem, Y. Boers, and H. Driessen, "Tracking closely spaced targets: Bayes outperformed by an approximation?," in *Proc. 11th Int. Conf. Inf. Fusion (FUSION 2008)*, Jun. 2008, pp. 1–8.

[1] MATLAB source code is available at http://www.cloudrunner.eu

[9] Y. Boers, E. Sviestins, and H. Driessen, "Mixed labelling in multitarget particle filtering," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 46, no. 2, pp. 792–802, 2010.

[10] D. F. Crouse, P. Willett, and Y. Bar-Shalom, "Developing a real-time track display that operators do not hate," *IEEE Trans. Signal Process.*, vol. 59, no. 7, pp. 3441–3447, Jul. 2011.

[11] D. F. Crouse, P. Willett, and Y. Bar-Shalom, "Generalizations of Blom and Bloem's PDF decomposition for permutation-invariant estimation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, May 2011, pp. 3840–3843.

[12] D. F. Crouse, P. Willett, Y. Bar-Shalom, and L. Svensson, "Aspects of MMOSPA estimation," in *Proc. 50th IEEE Conf. Decision Control-Eur. Control Conf.*, Orlando, FL, USA, Dec. 2011, pp. 6001–6006.

[13] D. F. Crouse, P. Willett, L. Svensson, and Y. Bar-Shalom, "Comparison of compressed sensing, ML, and MMOSPA estimation for radar superresolution," in *Proc. 45th Asilomar Conf. Signals, Syst., Comput.*, 2011, pp. 1484–1490.

[14] D. F. Crouse, "Advances in displaying uncertain estimates of multiple targets," in *Proc. SPIE-Signal Process., Sensor Fusion, Target Recognit. XXII*, 2013, vol. 8745, pp. 874 504–874 504-31.

[15] D. F. Crouse, P. Willett, L. Svensson, D. Svensson, and M. Guerriero, "The set MHT," in *Proc. 14th Int. Conf. Inf. Fusion (FUSION 2011)*, Jul. 2011, pp. 1–8.

[16] D. F. Crouse, P. Willett, M. Guerriero, and L. Svensson, "An approximate minimum MOSPA estimator," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, May 2011, pp. 3644–3647.

[17] M. Cuturi and A. Doucet, "Fast computation of Wasserstein barycenters," in *Proc. 31st Int. Conf. Mach. Learn. (ICML-14)*, Oct. 2014, pp. 685–693.

[18] H. Edelsbrunner and L. J. Guibas, "Topologically sweeping an arrangement," *J. Comput. Syst. Sci.*, vol. 38, no. 1, pp. 165–194, 1989.

[19] H. Edelsbrunner, J. O'Rourke, and R. Seidel, "Constructing arrangements of lines and hyperplanes with applications," in *Proc. 24th Annu. Symp. Found. Comput. Sci.*, 1983, pp. 83–91.

[20] A. Garcia-Fernandez, M. Morelande, and J. Grajal, "Particle filter for extracting target label information when targets move in close proximity," in *Proc. 14th Int. Conf. Inf. Fusion (FUSION 2011)*, 2011, pp. 1–8.

[21] R. Georgescu, P. Willett, L. Svensson, and M. Morelande, "Two linear complexity particle filters capable of maintaining target label probabilities for targets in close proximity," in *Proc. 15th Int. Conf. Inf. Fusion (FUSION 2012)*, 2012, pp. 2370–2377.

[22] T. Gerstner, "Sparse grid quadrature methods for computational finance," Habilitation thesis, Univ. of Bonn, Bonn, Germany, 2007.

[23] M. Guerriero, L. Svensson, D. Svensson, and P. Willett, "Shooting two birds with two bullets: How to find minimum mean OSPA estimates," in *Proc. 13th Int. Conf. Inf. Fusion (FUSION 2010)*, 2010, pp. 1–8.

[24] J. Rabin, G. Peyr, J. Delon, and M. Bernot, "Wasserstein barycenter and its application to texture mixing," in *Scale Space and Variational Methods in Computer Vision*, ser. Lecture Notes in Computer Science. Berlin, Germany: Springer, 2012, vol. 6667, pp. 435–446.

[25] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Norwood, MA, USA: Artech House, 2004.

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

[27] R. P. Stanley, "Hyperplane arrangements," in *Geometric Combinatorics*, ser. IAS/Park City Mathematics Series, E. Miller, V. Reiner, and B. Sturmfels, Eds. Providence, RI, USA: Ame. Math. Soc., 2007, vol. 13, pp. 389–496.

[28] L. Svensson, D. Svensson, M. Guerriero, and P. Willett, "Set JPDA filter for multi-target tracking," *IEEE Trans. Signal Process.*, vol. 59, no. 10, pp. 4677–4691, 2011.

[29] J. Vermaak, S. Godsill, and P. Perez, "Monte Carlo filtering for multi target tracking and data association," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 41, no. 1, pp. 309–332, Jan. 2005.

[30] M. Werman, S. Peleg, and A. Rosenfeld, "A distance metric for multidimensional histograms," *Comput. Vis., Graph., Image Process.*, vol. 32, no. 3, pp. 328–336, 1985.

Marcus Baum received the Dipl.-Inform. degree in computer science from the University of Karlsruhe (TH), Germany, in 2007, and graduated as Dr.-Ing. (Doctor of Engineering) at the Karlsruhe Institute of Technology (KIT), Germany, in January 2013. From 2013 to 2014, he was postdoc and assistant research professor at the University of Connecticut, CT, USA. Currently, he is a senior researcher at the Karlsruhe Institute of Technology (KIT), Germany. His research interests are in the field of extended object tracking, multi-target tracking, nonlinear estimation, and sensor data fusion. Marcus Baum is associate administrative editor of the "Journal of Advances in Information Fusion (JAIF)" and serves as local arrangement chair of the "19th International Conference on Information Fusion (FUSION 2016)".

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 also 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 of IEEE Signal Processing Letters. He was editor-in-chief for IEEE Transactions on Aerospace and Electronic Systems (2006–2011), and was Vice President for Publications for AESS (2012–2014). He is a member of the IEEE AESS Board of Governors 2003–2009, 2011 to present. He was General Co-Chair (with Stefano Coraluppi) for the 2006 ISIF/IEEE Fusion Conference in Florence, Italy and for the 2008 ISIF/IEEE Fusion Conference in Cologne, Germany, Program Co-Chair (with Eugene Santos) for the 2003 IEEE Conference on Systems, Man & Cybernetics in Washington DC, and Program Co-Chair (with Pramod Varshney) for the 1999 Fusion Conference in Sunnyvale.

Uwe D. Hanebeck obtained his Ph.D. degree in 1997 and his habilitation degree in 2003, both in electrical engineering from the Technical University in Munich, Germany. He is a chaired professor of Computer Science at the Karlsruhe Institute of Technology (KIT) in Germany and director of the Intelligent Sensor-Actuator-Systems Laboratory (ISAS). Since 2005, he is the chairman of the Research Training Group RTG 1194 "Self-Organizing Sensor-Actuator-Networks" financed by the German Research Foundation.

His research interests are in the areas of information fusion, nonlinear state estimation, stochastic modeling, system identification, and control with a strong emphasis on theory-driven approaches based on stochastic system theory and uncertainty models. Research results are applied to various application topics like localization, human-robot-interaction, assistive systems, sensor-actuator-networks, medical engineering, distributed measuring system, and extended range telepresence.

He was the General Chair of the "2006 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2006)", Program Co-Chair of the "11th International Conference on Information Fusion (Fusion 2008)", Program Co-Chair of the "2008 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2008)", Regional Program Co-Chair for Europe for the "2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2010)", and will be General Chair of the "19th International Conference on Information Fusion (Fusion 2016)". He is a Member of the Board of Directors of the International Society of Information Fusion (ISIF), Editor-in-chief of its *Journal of Advances in Information Fusion* (JAIF), and associate editor for the letter category of the IEEE Transactions on Aerospace and Electronic Systems (TAES). He is author and coauthor of more than 300 publications in various high-ranking journals and conferences.