Property Management in Wireless Sensor Networks with Overcomplete Radon Bases

XIAOYE JIANG, Stanford University
MO LI, Nanyang Technological University
YUAN YAO, Peking University
LEONIDAS GUIBAS, Stanford University

This article presents a scalable algorithm for managing property information about moving objects tracked by a sensor network. Property information is obtained via distributed sensor observations, but will be corrupted when objects mix up with each other. The association between properties and objects then becomes ambiguous. We build a novel representation framework, exploiting an overcomplete Radon basis dictionary to model property uncertainty in such circumstances. By making use of the combinatorial structure of the basis design and sparse representations we can efficiently approximate the underlying probability distribution of the association between target properties and tracks, overcoming the exponential space that would otherwise be required. Based on the proposed theories, we design a fully distributed algorithm on wireless sensor networks. We conduct comparative simulations and the results validate the effectiveness of our approach.

Categories and Subject Descriptors: C.2 [**Computer Communication Networks**]: Distributed Systems; G.2.1 [**Discrete Mathematics**]: Combinatorics—*Permutations and Combinations*

General Terms: Theory, Algorithms

Additional Key Words and Phrases: Wireless sensor networks, property management, homogeneous spaces

ACM Reference Format:
Jiang, X., Li, M., Yao, Y., and Guibas, L. 2013. Property management in wireless sensor networks with overcomplete radon bases. ACM Trans. Sensor Netw. 9, 3, Article 36 (May 2013), 26 pages.
DOI: http://dx.doi.org/10.1145/2480730.2480739

1. INTRODUCTION

In this article, we address the *property management problem* in wireless sensor networks, which is related to the *identity management problem* [Guibas 2008]. Both problems arise in the context of accurately tracking and identifying multiple moving targets with distributed sensors in the field.

X. Jiang and L. Guibas acknowledge the support of ARO grants W911NF-10-1-0037 and W911NF-07-2-0027, as well as NSF grant CCF-1011228 and a gift from the Google Corporation. M. Li acknowledges support from COE SUG/RSS 20Aug201013/14 in Nanyang Technological University of Singapore. Y. Yao acknowledges support from the National Basic Research Program of China (973 Program 2011CB809105) NSFC (61071157), Microsoft Research Asia, and a professorship in the Hundred Talents Program at Peking University.
Authors' addresses: X. Jiang (corresponding author), Institute for Computational and Mathematical Engineering, Stanford University, CA; email: xiaoyej@standford.edu; M. Li, Department of Computer Science, Nanyang Technological University, P.R. China; Y. Yao, Department of Statistics, Peking University, P.R. China; L. Guibas, Department of Computer Science, Stanford University, CA.

DOI: http://dx.doi.org/10.1145/2480730.2480739

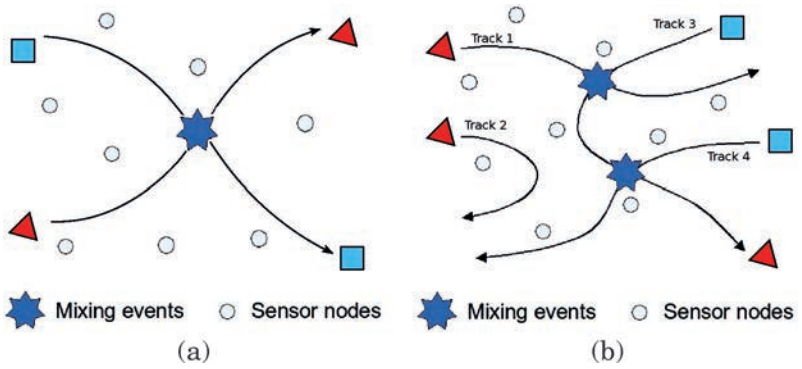


Fig. 1. Targets move in a sensor network. (a) Identities of targets can get confused when they pass each other. (b) Four red or blue targets (denoted by red triangles and blue squares) moving around. We get confused about colors when track 1 crosses with track 3, then track 3 crosses with track 4. In the end, we observe that the target on track 3 is red.

We first give a brief review of the identity management problem. Identity management for moving targets in the sensor networks was first introduced by Shin et al. [2003]. Suppose we have a sensor network which is tasked at tracking multiple, simultaneously moving targets in the monitored area. In such a setting, we assume that positions of the targets can be instantly tracked by the sensor nodes and we focus on distinguishing their identities during the tracking. When targets are well separated and good-quality observations are obtained by the sensors, the problem factorizes nicely. Different sensor nodes can focus on different targets, forming collaboration groups to best determine target identities. The problem becomes more complicated, however, when two targets come close to each other, which leads to confusion as the signal signatures of two targets mix up. After the two targets separate again, their positions may become immediately distinguishable, but their identities can still be confused, and the sensors may no longer be able to tell who is who.

Such uncertainties about identities will be carried forward in time with each target, until good-quality observations on their identities are obtained to allow disambiguation. How to achieve accurate and efficient disambiguation is subtle, for example, when the identity of target A becomes clear due to a new observation from a sensor close to A, another target B which A has mixed up with earlier becomes unambiguous as well; see Figure 1(a). Thus when there are many moving targets with mixed trajectories, it becomes increasingly complicated for the sensor network to resolve such ambiguities globally. Such a problem is called the identity management problem and the major task in addressing the problem is to maintain a belief state for the correct association between target tracks and target identities with continuous input of target mixing events and updated identity observations from sensors. The identity management problem poses a challenge for probabilistic inference as it needs to address the fundamental combinatorial challenge that there are a factorial number of possible associations to maintain between tracks and identities. There have been many works proposed to address the identity management problem.

Property management problem is another interesting problem related to the identity management problem. We note that in many cases, we do not need to distinguish the identity of every individual target or the sensors are not powerful enough to capture all target features for identification. Instead we may only get coarse property information on targets and such information will suffice for many applications. For example, when we track troops of different parties on the battlefield so as to infer which party the

individuals are affiliated with, it would suffice that we correctly track their affiliations rather than their identities. Here their affiliations are regarded as properties associated with the targets. As a matter of fact, tracking such properties associated with targets is a more natural assumption for sensor networks because what sensors directly sense is property information rather than identity features. While there have been many efforts put in studying the identity management problem, few studies have been done for the property management problem in sensor networks.

In this article, we conduct the first study on such a problem, where we focus on target properties rather than their identities. For simplicity of exposition, we restrict our attention to the basic case where the targets can be classified into two categories, red or blue. Similar to the identity management problem, confusion will arise when a red target and a blue target mix up with each other and then depart. Future property observations by sensors can help to resolve such ambiguities. The property management problem is to maintain a belief state for the correct association between target tracks and target properties (red or blue) with target mixing events and updated property observations.

In the setting where there are k red targets and $n - k$ blue targets, there would be $\binom{n}{k}$ total possibilities about which k tracks contain the red targets. To overcome such a combinatorial complexity challenge of storing a distribution of length $\binom{n}{k}$, we introduce the use of hierarchical *Radon bases* dictionary. It turns out that each Radon basis can be indexed by a discrete set, which allows us to implicitly characterize the distribution by storing only a collection of discrete sets together with coefficients. Moreover, we can update the distribution by updating the discrete sets and coefficients which reflect the happening of mixing and observation events. In addition, all the Radon bases form an overcomplete dictionary, which makes it possible for us to explore sparse approximations of a distribution. Thus, we are able to always maintain a compact summary about which k tracks contain the red targets with the Radon bases machinery.

The property management problem is closely related to the identity management problem; we can reveal target identities if we have enough target properties to differentiate the identities. Typically, $\mathcal{O}(\log n)$ target properties are adequate to completely identify an object out of n. Thus, property management provides us an alternative method to study the identity management problem, while the identity management solutions cannot be applied for property management. The property management problem has to be addressed separately, as multiple targets may share the same property and the permutation machinery assumption that each track corresponds to a unique identity no longer applies.

We summarize the contributions of our article as follows.

(1) To the best of our knowledge, this is the first work to address the property management problem with sensor networks.
(2) We use novel overcomplete bases together with sparse approximation algorithms to represent uncertainties to achieve high accuracy as well as low computational and communication overhead.
(3) We propose a fully distributed algorithm which is easy to implement and lightweight for sensor network processing.

The article is organized as follows: In Section 2, we introduce some related works. In Section 3, we use a Markov model to formulate the property management problem. A novel overcomplete Radon basis dictionary for representing uncertainties over homogeneous spaces is introduced in Section 4. In Section 5, scalable algorithms are provided based on the proposed framework. We conduct comparative simulations to validate this approach in Section 6. In Section 7, we conclude this article.

2. RELATED WORKS

The key computational challenge in the identity management problem is that the number of possible associations between tracks and target identities can be very large. To address such a problem, Shin et al. [2003] use the *belief matrix* to approximate the association probabilities, which collapses the factorial distribution to its first-order marginals (marginal probability that identity i is associated with track j). An alternative representation [Schumitsch et al. 2005] is using an *information matrix* whose elements represent marginal log-likelihoods. Both methods provide efficient and scalable algorithms yet fail to characterize higher-order marginals, such as the association probabilities between pairs of tracks and pairs of identities. The marginals of different orders are interconnected, thus the formulation becomes quickly unmanageable. Fortunately, there is an established mathematical theory that ideally suits to disentangling all the information: the *representation theory of permutation group* [Diaconis 1988; Sagan 2001; Serre 1977]. It turns out that one can define Fourier transforms for functions over all *permutations*, and low (high)-order Fourier coefficients contain information about low (high)-order marginals. Kondor et al. [2007] use a general set of Fourier coefficients to represent uncertainty over permutations and demonstrate improvements against only using low-order Fourier coefficients. Recently, Huang et al. [2009a] proposed an algorithm, called *Kronecker Conditioning*, which performs all probabilistic inference operations completely in the Fourier domain. Such a method can address any mixing or observation model and gains efficiency by truncating the Fourier expansions, allowing for a principled trade-off between computational complexity and approximation accuracy. Though polynomial, the Fourier methods are still quite computationally demanding when the number of targets is of even modest size. One way to mitigate the overhead is to factorize the problem into smaller clusters, so that highly certain individual or group associations can be pulled out of a global Fourier representation and represented compactly [Huang et al. 2009b].

The identity management problem is not identical with the classical data association problem of maintaining correspondences between tracks and observations. In the identity management problem, the rate at which observations happen that are informative about target identities is not coupled to the rate of observations about target positions and can be much lower. We note that a vast literatures already exist on the the data association problem, beginning with the Multiple Hypothesis Testing approach (MHT) of Reid [1979]. MHT is a deferred logic method in which past observations are exploited in forming new hypotheses when a new set of observations arises. Since the number of hypotheses can grow exponentially over time, various heuristics have been proposed to help cope with the complexity. For example, one can choose to maintain only the k best hypotheses for some parameter k [Cox and Hingorani 1994], using Murty's algorithm [Murty 1968]. But for the approximation to be effective, k still has to be exponential in the number of targets. A slightly more recent filtering approach is the Joint Probabilistic Data Association filter (JPDA) [Bar-Shalom and Fortmann 1988], which is a suboptimal single-stage approximation of the optimal Bayesian filter. JPDA makes associations sequentially and is unable to correct erroneous associations made in the past [Poore 1995]. Even though it is more efficient than MHT, the calculation of the JPDA association probabilities is still an NP-hard problem [Collins and Uhlmann 1992]. Polynomial approximation algorithms to the JPDA association probabilities have recently been studied using Markov Chain Monte Carlo (MCMC) methods [Oh et al. 2004; Oh and Sastry 2005]. Generalized sensor models, for example, binary sensors [Aslam et al. 2003] which can only tell whether the target is moving toward the sensor or away from the sensor, as well as the related localizability problems [Yang et al. 2010] have also been considered in the literature. However, none of

Table I. Localized Mixing and Observation Data

Event #	Event Type
1	Tracks 1 and 3 mixed
2	Tracks 3 and 4 mixed
3	Observed target on track 3 is red

these proposed approaches can be used to address the property management problem. To the best of our knowledge, there is neither theoretical study on the problem itself nor algorithmic efforts in making distributed solutions in sensor networks.

Modern computer technologies have made it possible for us to deploy densely distributed sensor network systems. Such systems can hold up to hundreds of sensor nodes, which can perform lots of sensing and controling tasks such as multitarget tracking [Dutta et al. 2006; Liu et al. 2011; Zhu et al. 2012], intrusion detection [Arora et al. 2004], ecosystem surveillance [Mo et al. 2009], etc. The classical problem of reliably tracking also connects the vision community if one considers the camera network for object detection and recognition [Xie et al. 2008].

Given that the tracking literature is becoming mature, however, we note that the property management problem still needs to be addressed. In this problem, we don't assume a dense in time and accurate in space measurements of the target positions which are typically assumed in the tracking literature because these dense measurements are quite expensive to acquire. We also note that in a sensor network, often different types of sensors convey property information rather than positional information, and typically the former are more expensive. For example, we can imagine a network of simple proximity sensors that can be used to detect the presence of targets (and therefore provide information about locations), but which cannot differentiate the mobile targets from each other. These inexpensive sensors then can be augmented with a network of sparse but expensive camera sensors that can observe other properties of targets, such as colors, which help in differentiating or identifying the targets. The latter observations occur much less frequently, however. Thus, such a problem setup deserves research attention from a theoretical perspective.

3. FORMULATION

We start with a simple tracking problem with four target tracks. As depicted in Figure 1(b), four targets, where two are red and the other two are blue, are moving within a field deployed with sensors. The sensors are capable of sensing target properties (red or blue). As we have mentioned earlier, the property management problem requires to maintain a belief state for the correct association between target tracks and target properties (red or blue), without distinguishing among red (blue) targets.

In this particular example, when the four targets are moving within the field, local sensors may report two types of events, namely, *mixing events* (i.e., two tracks get mixed when the targets get too close to each other) and *observation events* (i.e., the target property on a particular track is clearly observed by a local sensor). Hence, a stream of localized data is observed about the four tracks, which is recorded in Table I.

Assume initial colors of the targets are known. Then from Table I, we know track 2 never mixes with other tracks. Observing red target on track 3 will clarify all the ambiguities, for example, targets on tracks $\{2, 3\}$ are red and targets on tracks $\{1, 4\}$ are blue. Such a simple example illustrates the combinatorial nature of the property management problem; reasoning on the mixing events allows us to determine which targets move along which tracks even though we only have partial observations on the tracks.

In the following, we introduce a Markov model to formulate the property management problem.

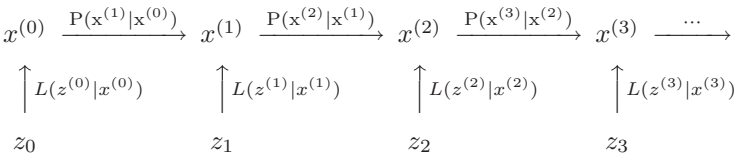


Fig. 2. The Markov model for the property management problem.

3.1. The Markov Model

Consider we have n targets, k of them are red and $n-k$ of them are blue. We consider all possible k-sets of the set $\{1, 2, \ldots, n\}$, so that each k-set characterizes a state indicating which k tracks have the red targets. We introduce a *Markov model* to model the uncertainty, which is represented by a probability distribution f over all k-sets $x^{(t)}$ at time $t = 0, 1, 2, \ldots$. Such a distribution encodes the probability of an arbitrary k-set of the tracks being red at time t. As Figure 2 illustrates, we will update the distribution f over $x^{(t)}$ at each time step.

In Figure 2, to model the conditional probability distribution $P(x^{(t)}|x^{(t-1)})$, we will work on a *mixing model* so that the mixing model reflects, for example, that the targets belonging to two tracks are swapped with some probability at a mixing event. To model the distribution $L(z^{(t)}|x^{(t)})$, we will work on an *observation model*, which captures the likelihood of observation $z^{(t)}$, given that targets on a k-set of tracks $x^{(t)}$ are all red.

We focus on *filtering*, where one queries the Markov model for posterior at each time step, based on all past observations. Given distribution $f(x^{(t)}|z^{(0)}, \ldots, z^{(t)})$, we recursively compute $f(x^{(t+1)}|z^{(0)}, \ldots, z^{(t+1)})$ with two steps: a *rollup step* and a *conditioning step*. The rollup step multiplies the distribution in the mixing model and the distribution of the previous step.

$$f\left(x^{(t+1)}|z^{(0)}, \ldots, z^{(t)}\right) = \sum_{x^{(t)}} P\left(x^{(t+1)}|x^{(t)}\right) f\left(x^{(t)}|z^{(0)}, \ldots, z^{(t)}\right)$$

The conditioning step conditions the distribution on an observation $z^{(t+1)}$ using Bayes rule.

$$f\left(x^{(t+1)}|z^{(0)}, \ldots, z^{(t+1)}\right) \propto L\left(z^{(t+1)}|x^{(t+1)}\right) f\left(x^{(t+1)}|z^{(0)}, \ldots, z^{(t)}\right)$$

Since the space of all k-sets of an n-set is of size $\binom{n}{k}$, a single iteration of the algorithm requires at least $\mathcal{O}(\binom{n}{k})$ operations, which is not polynomial with k. As will be detailed later, the approach that we advocate is to use a novel representation of distributions over all k-sets, through which we can always maintain a compact representation for an arbitrary distribution. We will also present scalable algorithms such that updating distributions under the rollup and conditioning steps in such a representation framework requires polynomial complexity in both n and k.

The unique feature of this problem is that there are inherent dependencies among the property values, where the joint probability cannot be factorized as products of marginal probabilities. Many typical compact representations, such as graphical models, cannot capture the inherent mutual exclusivity constraints associated with the problem where k targets are red reflecting that the other $n-k$ targets are blue.

3.2. Permutation Groups and Homogeneous Spaces

In this section, we formally introduce *permutation groups* and *homogeneous spaces* which are used in modeling our problem. In mathematics, the set of all k-sets of an

n-set is known as a *homogeneous space*. Such a homogeneous space is associated with the *permutation group*, which acts on the homogeneous space in a transitive way.

Definition 1. A permutation on n elements is a one-to-one mapping of the set $\{1, \ldots, n\}$ onto itself and can be written as a tuple, $\sigma = [\sigma(1), \sigma(2), \ldots, \sigma(n)]$, where $\sigma(i)$ denotes where the i-th element is mapped with the permutation. The set of all permutations on n elements forms the *permutation group* S_n under the operation of function composition. We sometimes notate $\sigma = (i, j)$ which denotes a swap of i with j.

Definition 2. The collection of all k-sets of $\{1, 2, \ldots, n\}$ is a *homogeneous space*, denoted by X^k. The permutation group S_n acts on X^k in the following way: suppose $\sigma \in S_n$ and $x = \{x_1, x_2, \ldots, x_k\} \in X^k$, then $\sigma x = \sigma\{x_1, x_2, \ldots, x_k\} = \{\sigma(x_1), \sigma(x_2), \ldots, \sigma(x_k)\}$. It is easy to verify that S_n acts transitively on X^k(any $x, y \in X^k$ there exists a σ such that $\sigma x = y$).

A permutation acting on a homogeneous space models the process of how tracks followed by targets might be mixed or swapped. Recently, Jiang et al. [2009] invented an approach based on the homogeneous space to study the clique detection problem in social networks. Although homogeneous spaces are well-studied objects in mathematics, to the best of our knowledge, they have not been used before to model such association problems.

Example 3.1. In the previous example depicted in Figure 1(b), we consider the homogeneous space: all 2-sets of $\{1, 2, 3, 4\}$ (denoting the four tracks). At the beginning, targets on tracks $\{1, 2\}$ are red. If tracks 1 and 3 swapped the targets, then targets on tracks $\{2, 3\}$ will be red. Using mathematical terms, it can be stated in the following way: the permutation $(1, 3)$ acts on $\{1, 2\}$ (an element in X^2) will be $\{2, 3\}$.

3.3. Mixing Model

In this article, we consider a particular class of probabilistic mixing models: that of random walks over the permutation group, which assumes that $x^{(t+1)}$ is generated from $x^{(t)}$ by drawing a random permutation $\sigma^{(t)}$ from some distribution $Q^{(t)}$ over the permutation group S_n. With such a probabilistic mixing model, we can write the rollup operation as a Markov transition matrix times the prior distribution. In our problem, $\sigma^{(t)} \in S_n$ represents a random permutation that might occur among tracks when they get too close to each other. As we have introduced in the previous section, a permutation $\sigma^{(t)}$ acts on a state $x^{(t)}$ in the homogeneous space X^k as $\sigma^{(t)}x^{(t)}$. Hence, the distribution over $x^{(t+1)}$ generated from $x^{(t)}$ by a random draw from the distribution Q over S_n is

$$f\left(x^{(t+1)}|z^{(0)}, \ldots, z^{(t)}\right) = \sum_{x^{(t)}} P\left(x^{(t+1)}|x^{(t)}\right) f\left(x^{(t)}|z^{(0)}, \ldots, z^{(t)}\right)$$

$$= \sum_{(x^{(t)},\sigma):\ \sigma x^{(t)} = x^{(t+1)}} Q(\sigma) f\left(x^{(t)}|z^{(0)}, \ldots, z^{(t)}\right) = T\left(x^{(t)}, x^{(t+1)}\right) f\left(x^{(t)}|z^{(0)}, \ldots, z^{(t)}\right),$$

where $T(x, y) = \sum_{\sigma:\sigma x = y} Q(\sigma)$, meaning that all $Q(\sigma)$ such that $\sigma x = y$ will contribute to the $(x, y)-$entry of the transition matrix. In addition, we have the following theorem.

THEOREM 1. *Let Q be a probability distribution on S_n, then Q induces a doubly stochastic Markov transition matrix for X^k with transitions: $T(x, y) = \sum_{\sigma:\sigma x = y} Q(\sigma)$.*

The preceding theorem [Diaconis 1988] gives an explicit formula for transition matrices of a distribution over the homogeneous space. As we will see later, transition matrices induced from the distribution Q also interact nicely with the homogeneous space, which can be utilized to simplify the computation of the rollup step.

Table II. Updated Priors when Mixing Happens

	$\{1,2\}$	$\{1,3\}$	$\{1,4\}$	$\{2,3\}$	$\{2,4\}$	$\{3,4\}$
$f(x^{(0)})$	1	0	0	0	0	0
$f(x^{(1)})$	1/2	0	0	1/2	0	0
$f(x^{(2)})$	1/2	0	0	1/4	1/4	0

In this article, we consider the simplest probabilistic mixing model which assumes that with probability p, nothing happens to the two targets, and with probability $(1-p)$, the targets for tracks i and j are swapped (similar models are considered in Huang et al. [2009a]). The probability distribution Q over S_n for this probabilistic mixing model is therefore

$$Q(\sigma) = \begin{cases} p & \text{if } \sigma = id \\ 1-p & \text{if } \sigma = (i,j) \\ 0 & \text{otherwise} \end{cases}.$$

We note that there are special structures that we can explore in the transition matrix induced from the particular distribution Q over S_n, that is, each row or column has either one nonzero entry (which must be 1) or two nonzero entries (which must be p and $1-p$). We will use such a fact to do the rollup operation in an efficient way.

Example 3.2. We run the mixing update routines on the first two mixing events of the example in Figure 1(b). For each mixing event, we assume two tracks i and j swap targets with equal probability. Using the probabilistic mixing model we obtain distributions $f(x^{(t)})$ for $t = 0, 1, 2$ as shown in Table II. Here, $f(x^{(t)}|z^{(0)}, z^{(1)}, \ldots, z^{(t)})$ is abbreviated as $f(x^{(t)})$.

3.4. Observation Model

In contrast to the rollup step, the conditioning step can potentially decrease uncertainty. We use Bayes rules to find the posterior distribution $P(x^{(t)}|z^{(t)})$ after observing some evidence $z^{(t)}$, which can be expressed as the following.

$$f\left(x^{(t)}|z^{(0)}, z^{(1)}, \ldots, z^{(t)}\right) = \frac{L\left(z^{(t)}|x^{(t)}\right) f\left(x^{(t)}|z^{(0)}, z^{(1)}, \ldots, z^{(t)}\right)}{\sum_{x^{(t)}} L\left(z^{(t)}|x^{(t)}\right) f\left(x^{(t)}|z^{(0)}, z^{(1)}, \ldots, z^{(t)}\right)}$$

It requires two steps to compute the posterior: a *pointwise product* of prior $f(x^{(t)}|z^{(0)}, z^{(1)}, \ldots, z^{(t)})$ and likelihood $L(z^{(t)}|x^{(t)})$, followed by a normalization step, which is computing $\sum_{x^{(t)}} L(z^{(t)}|x^{(t)}) f(x^{(t)}|z^{(0)}, z^{(1)}, \ldots, z^{(t)})$.

The simplest observation model assumes that we get observation z of the form: "see red on track i" (similar models are considered in Huang et al. [2009a] and Kondor et al. [2007]). Now we assume all red (blue) targets have the same color histograms; sensors sense properties by a random draw from the color histogram of the target. If, for example, all red targets have 80% of red, 10% of blue, and 10% of other colors (yellow, grey, etc.) while all blue targets have 70% of blue, 20% of red, and 10% of other colors, then the likelihood function for observation event z = "see red on track i" given "targets on k-set tracks x are red" is:

—if $i \in x$, $L(z|x) = L(z|x \text{ are red}) = L(z|i \text{ is red}) = .8$;
—if $i \notin x$, $L(z|x) = L(z|x \text{ are red}) = L(z|x^c \text{ are blue}) = L(z|i \text{ is blue}) = .2$.

So we have

$$L(z|x) = \begin{cases} .8 & \text{if } i \in x \\ .2 & \text{if } i \notin x \end{cases}.$$

We conclude this section by pointing out that both the rollup step and conditioning step are of complexity at least $\mathcal{O}(\binom{n}{k})$, if we explicitly form the distribution on the homogeneous space. In the next section, we will explore the math structure of homogeneous spaces to overcome such a complexity issue.

4. EFFICIENT REPRESENTATION

In this section, we propose a novel representation framework to characterize distributions over homogeneous spaces.

4.1. Hierarchical Radon Basis

Let $L(X^k)$ be the set of all functions on the homogeneous space X^k. There is a special technique for decomposing $L(X^k)$. For each $0 \leq j \leq k$, define a matrix $\Phi^{j,k}$ as an $\binom{n}{k}$ by $\binom{n}{j}$ matrix where each row represents a k-set and each column represents a j-set. The entries in $\Phi^{j,k}$ are binary, indicating whether the j-set is a subset of the k-set. The matrix $\Phi^{j,k}$ can be interpreted as a mapping from functions on all j-sets to functions on all k-sets. The columns of the matrix $\Phi^{j,k}$ span an $\binom{n}{j}$-dimensional subspace of $L(X^k)$, which are called the *Radon bases* of order j. With a Radon basis we can represent a distribution over k-sets by using coefficients on j-sets, where $0 \leq j \leq k$.

In Section 3.3, we noticed that a probability distribution Q over S_n can induce an action on a distribution over all k-sets, that is, Q induces a transition matrix for updating distributions over all k-sets. However, there is nothing special about k when we define the transition matrix there. Thus we can generally let Q induce a transition matrix for updating distributions over all j-sets, where $0 \leq j \leq k$. Here the transition matrix is $T^j(x, y) = \sum_{\sigma:\, \sigma x = y} Q(\sigma)$, where x, y are two j-sets. The following proposition summarizes important properties regarding $\Phi^{j,k}(0 \leq j \leq k)$ and the transition matrices induced from Q.

THEOREM 2. *Let Q be a distribution over S_n, we have the following three propositions:*

(1) $R(\Phi^{0,k}) \subset R(\Phi^{1,k}) \subset \cdots \subset R(\Phi^{k,k})$ *where $R(\cdot)$ denotes the range of a matrix.*
(2) *Each subspace $R(\Phi^{j,k})$ is invariant under the action of the distribution Q.*
(3) *The matrix $\Phi^{j,k}$ commutes with the action of the distribution Q, that is, $\Phi^{j,k}T^j = T^k\Phi^{j,k}$ where T^j, T^k are transition matrices induced from the same distribution Q.*

The first proposition states that the range of matrices $\Phi^{j,k}$ forms a hierarchical decomposition of $L(X^k)$. Bases that span $R(\Phi^{j,k})$ of small (large) j are efficient for approximating smooth (peaky) distributions over X^k respectively. We note that such a hierarchical basis design derives from the representation theory of permutation groups in such a way that each subspace is invariant under group actions [Diaconis 1988]. The second proposition tells us that if $f \in L(X^k)$ lies in the range of $\Phi^{j,k}$ for some $j \leq k$, then after the rollup operation, the updated distribution still lies in the range of $\Phi^{j,k}$. The third proposition suggests us an efficient algorithm to update f when $f \in R(\Phi^{j,k})$. Suppose we have a distribution over k-sets $f = \Phi^{j,k}c_j$ where c_j are coefficients on j-sets, then because of this commutative property, the rollup operation for f can be performed simply by computing rollup operation for coefficients c_j.

$$T^k f = T^k\Phi^{j,k}c_j = \Phi^{j,k}T^j c_j = \Phi^{j,k}(T^j c_j)$$

One interesting fact about the Radon basis matrix $\Phi^{j,k}$ is that it has a pseudo-inverse $(\Phi^{j,k})^+$ which maps from functions on k-sets to functions on j-sets. $\Phi^{j,k}$ and $(\Phi^{j,k})^+$ satisfy that $(\Phi^{j,k})^+\Phi^{j,k}$ is identity and $\Phi^{j,k}(\Phi^{j,k})^+$ is an orthogonal projection. Given Radon bases consisting of delta functions on all j-sets and k-sets, respectively,

the (r, s) element of $(\Phi^{j,k})^+$ is $\frac{(-1)^{k-j}(k-j)}{(-1)^{|s-r|}|s-r|\binom{n-j}{|s-r|}}$, where $s-r$ means the set difference, that is, $s-r = s \cap r^c$.

4.2. Overcomplete Basis Representation

Recent approaches [Huang et al. 2009a; Kondor et al. 2007] in modeling the identity management problem keep a compact representation of distributions over permutation groups by storing only low-order Fourier coefficients. Clearly, similar ideas can be adopted here; we can use coefficients for low-order Radon bases ($\Phi^{j,k}$ with small j) to represent a distribution over the homogeneous space X^k. Using a low-order Radon basis, however, fails to characterize highly certain cases, for example, a delta function on X^k which can be characterized by a single basis in $\Phi^{k,k}$, while a low-order Radon basis is incapable of representing accurately such a peaky distribution; on the other hand, high-order Radon bases are not efficient for representing smooth distributions over X^k, for example, a constant function can be characterized by a single basis in $\Phi^{0,k}$, while one cannot have a compact representation for such a smooth function by using high-order bases. Similar problems happen in the identity management problem, where a low-order Fourier basis fails to represent peaky functions on permutation groups while high-order Fourier bases are not efficient to represent smooth functions. In reaction to this, we propose to use an overcomplete Radon basis dictionary to represent distributions over homogeneous space X^k where we concatenate all $\Phi^{j,k}$'s, that is,

$$f = \Phi^{0,k}c_0 + \Phi^{1,k}c_1 + \cdots + \Phi^{k,k}c_k = \sum_{j=0}^{k} \Phi^{j,k}c_j,$$

where c_j are coefficients on j-sets. By using a hierarchical overcomplete Radon basis, we will hopefully have sparse representation for any distribution over X^k.

Example 4.1. In the example in Figure 1(b), the distribution f at $t = 2$ can be represented with coefficients $c_0 = 0$, $c_1 = [0, 1/4, 0, 0]^T$, and $c_2 = [1/4, 0, 0, 0, 0, 0]^T$ to f as the second column of $\Phi^{1,2}$ indicates three 2-sets which contain 2.

5. ALGORITHM DESIGN

In this section, we design algorithms based on aforementioned theorems for updating probabilistic distributions over the homogeneous space X^k. We assume that, using overcomplete bases, sparse representations are available for the distributions over X^k which we work with. By making use of the combinatorial structure of the basis matrices and our sparse representation assumptions, we obtain efficient algorithms for updating distributions over X^k polynomial in n and k, which will be detailed in Section 5.1 and Section 5.2. When we keep updating the Radon basis coefficients, however, we may gradually lose sparsity. To resolve such an issue, we propose in Section 5.3 sparse approximation algorithms to reorganize the coefficients and regain sparsity. To make our approach applicable in distributed sensor networks, we further develop distributive ways of doing all such algorithms within the sensor network in Section 5.4.

5.1. Algorithm for Rollup Step

Given a distribution f over homogeneous space X^k using overcomplete Radon bases, we have $f = \sum_{j=0}^{k} \Phi^{j,k}c_j$ where c_j are coefficients over j-sets. If each $c_j (0 \le j \le k)$ is sparse, then we can store f by only storing the nonzero values in each c_j and their corresponding column indices, which are j-sets. We can represent f in another way.

ALGORITHM 1: Algorithm for Rollup Step

Input: A collection of sets with associated values $I = \{(\alpha, c_\alpha) : c_\alpha \neq 0\}$
Output: A collection of sets with associated values O
Procedure:
Initialize $O \leftarrow \{\}$
for each $(\alpha, c_\alpha) \in I$ **do**
  if $t_i \in \alpha, t_j \in \alpha$ or $t_i \notin \alpha, t_j \notin \alpha$ **then**
    $O \leftarrow O \cup \{(\alpha, c_\alpha)\}$
  else if $t_i \in \alpha, t_j \notin \alpha$ or $t_i \notin \alpha, t_j \in \alpha$ **then**
    $\beta \leftarrow \alpha \Delta \{t_i, t_j\}$
    Retrieve value v_α associated with α, if α exist in O; otherwise set $v_\alpha = 0$
    Retrieve value v_β associated with β, if β exist in O; otherwise set $v_\beta = 0$
    $O \leftarrow O \cup \{\alpha, v_\alpha + pc_\alpha\}$
    $O \leftarrow O \cup \{\beta, v_\beta + (1-p)c_\beta\}$
  end if
end for

We have

$$f = \sum_\alpha \Phi_\alpha c_\alpha,$$

where α is a set of size at most k, and c_α is the coefficient for the basis column Φ_α.

We now describe the algorithm for updating Radon basis coefficients when mixing events happen. In particular, we consider probabilistic mixing models as described in Section 3.3. When a mixing event happens, we need to perform a rollup operation. We have $f \leftarrow T^k f$ where T^k is a transition matrix for distribution f over all k-sets, and T^k is induced from a probability Q on permutation group S_n. In Section 4.1, we know that the basis matrices $\phi^{j,k}$ commute with the action of a distribution Q over S_n. Hence, we have

$$T^k f = T^k \sum_{j=0}^{k} \Phi^{j,k} c_j = \sum_{j=0}^{k} T^k \Phi^{j,k} c_j = \sum_{j=0}^{k} \Phi^{j,k} T^j c_j.$$

To update f, we only need to update coefficients c_j's as $c_j \leftarrow T^j c_j$, for $0 \leq j \leq k$.

When tracks t_i and t_j mix, according to the probabilistic mixing model in Section 3.3, we have a distribution Q which takes nonzero values only on id and (t_i, t_j). Note that there is a special structure within the induced transition probability matrix; each column of the transition matrix T^j has either one nonzero entry (which is 1) or two nonzero entries (which are p and $1-p$). If we store f by using a collection of sets α's with associated values c_α's, then we can efficiently get the result of the rollup operation. In cases where $t_i \in \alpha, t_j \in \alpha$, or $t_i \notin \alpha, t_j \notin \alpha$, $\Phi_\alpha c_\alpha$ will not be affected after updating, that is, $T^k(\Phi_\alpha c_\alpha) = \Phi_\alpha c_\alpha$; in cases where $t_i \in \alpha, t_j \notin \alpha$, or $t_i \notin \alpha, t_j \in \alpha$, $\Phi_\alpha c_\alpha$ will be split into pc_α and $(1-p)c_\alpha$ on α and $\beta = \alpha \Delta \{t_i, t_j\}$, where Δ denotes the symmetric difference between two sets ($A \Delta B = (A \cup B) \cap (A \cap B)^c$), that is, $T^k(\Phi_\alpha c_\alpha) = \Phi_\alpha(pc_\alpha) + \Phi_\beta(1-p)c_\alpha$.

We have the following theorem regarding Algorithm 1.

THEOREM 3. *Suppose $f = \sum_\alpha \Phi_\alpha c_\alpha$, N is the number of nonzero coefficients c_α's and k is the number of red targets. The rollup algorithm can generate output in $\mathcal{O}(kN \log N)$ computational time. The size of nonzero coefficients in the output is at most $2N$.*

Example 5.1. In the example in Figure 1(b), at time $t = 1$ the distribution over 2-sets can be stored by

$$I = \{(\{1, 2\}, 1/2), (\{2, 3\}, 1/2)\}.$$

After the mixing event happens between tracks 3 and 4 where we have a probability distribution Q over S_4,

$$Q(\sigma) = \begin{cases} 1/2 & \text{if } \sigma = id \\ 1/2 & \text{if } \sigma = (3,4) \\ 0 & \text{otherwise} \end{cases}.$$

The updated distribution can be stored as

$$O = \{(\{1,2\}, 1/2), (\{2,3\}, 1/4), (\{2,4\}, 1/4)\}.$$

5.2. Algorithm for Conditioning Step

Two computation phases are involved in the conditioning step. First, a pointwise product needs to be computed, and second, we need to compute the normalizing constant. Note that in the observation model in Section 3.4, the likelihood functions $L(z|x)$ are of the form

$$L(z|x) = \begin{cases} a & \text{if } i \in x \\ b & \text{if } i \notin x \end{cases}.$$

Such a likelihood function $L(z|x)$ lies in the space of $R(\Phi^{1,k})$. If we define a function L_1 on 1-set as

$$L_1(j) = \begin{cases} a - \frac{k-1}{k} b & \text{if } j = i \\ \frac{1}{k} b & \text{if } j \neq i \end{cases},$$

it is easy to verify that $L(z|x) = R^{1,k} L_1$. We can even express $L(z|x)$ in a more compact way: $L(z|x) = b\Phi_\emptyset + (a-b)\Phi_{\{i\}}$.

Given $f(x) = \sum_\alpha \Phi_\alpha c_\alpha$ and $L(z|x) = \sum_\beta \Phi_\beta c_\beta$ where β's are at most 1-sets, the pointwise product is

$$f(x) \cdot L(z|x) = \left(\sum_\alpha \Phi_\alpha c_\alpha \right) \cdot \left(\sum_\beta \Phi_\beta l_\beta \right) = \sum_{\alpha,\beta} c_\alpha l_\beta (\Phi_\alpha \cdot \Phi_\beta),$$

where the last equality is due to the distributive law for the pointwise product operation.

For the basis vector corresponding to α and the basis vector corresponding to β, the pointwise product between Φ_α and Φ_β can be estimated as

$$\Phi_\alpha \cdot \Phi_\beta = \begin{cases} \Phi_{\alpha \cup \beta} & \text{if } |\alpha \cup \beta| \leq k \\ \mathbf{0} & \text{if } |\alpha \cup \beta| > k \end{cases}.$$

The normalizing constant actually equals the l_1 norm of $f(x) \cdot L(z|x)$. If we have $f(x) \cdot L(z|x) = \sum_\gamma \Phi_\gamma c_\gamma$, then $Z = \sum_\gamma |\Phi_\gamma|_1 c_\gamma = \sum_\gamma \binom{n-|\gamma|}{k-|\gamma|} c_\gamma$.

The algorithm for updating Radon basis coefficients when an observation event happens is summarized in Algorithm 2, which can be used to deal with general likelihood functions. In the special case where the likelihood function can be compactly represented as a linear combination of $\Phi_\emptyset$ and $\Phi_{\{i\}}$, we have the following theorem.

THEOREM 4. *Suppose $f = \sum_\alpha \Phi_\alpha c_\alpha$, N is the number of nonzero coefficients c_α's and k is the number of red targets. The conditioning algorithm can generate output in $\mathcal{O}(kN \log N)$ computational time. The size of nonzero coefficients in the output is at most $2N$.*

We remark that in the conditioning algorithm, the normalizing step is not essential because if we do not normalize the distribution, the result is still accurate

ALGORITHM 2: Algorithm for Conditioning Step — Computing Posterior Distribution

Input: Two collection of sets with associated values $I_1 = \{(\alpha, c_\alpha) : c_\alpha \neq 0\}$, $I_2 = \{(\beta, l_\beta) : l_\beta \neq 0\}$
Output: A collection of sets with associated values O
Procedure:
for each $(\alpha, c_\alpha) \in I_1$ **do**
  for each $(\beta, l_\beta) \in I_2$ **do**
    if $|\alpha \cup \beta| \leq k$ **then**
      Retrieve value v_γ associated with γ, if γ exist in O; otherwise set $v_\gamma = 0$
      $\gamma \leftarrow \alpha \cup \beta$
      $v_\gamma \leftarrow v_\gamma + c_\alpha l_\beta$
      $O \leftarrow O \cup \{(\gamma, v_\gamma)\}$
    end if
  end for
end for
Compute normalizing constant $Z \leftarrow \sum_\gamma \binom{n-|\gamma|}{k-|\gamma|} c_\gamma$
Divide each c_γ in O by Z

up to a multiplication constant. We may even choose a special likelihood function $L(z|x) = b\Phi_\emptyset + (a-b)\Phi_{\{i\}}$ where $b = 1$ with the benefit that if $f = \sum_\alpha \Phi_\alpha c_\alpha$ is the prior distribution, then c_α will not change after the conditioning step when the reported target i does not belong to α. Such a technique will be useful in Section 5.4.

5.3. Sparse Approximation

We have developed algorithms for updating Radon basis coefficients at the rollup step and conditioning step. For the rollup step, coefficients only propagate within Radon bases of the same order while for the conditioning step coefficients may propagate to Radon bases of higher orders, that is, the more observations we have, the more we will be certain about which k-set has the red targets.

The one-step rollup and conditioning algorithms are quite scalable. As we keep updating the Radon basis coefficients using the rollup and conditioning algorithms, however, we may need more and more coefficients to represent the distribution over all k-sets (as revealed by Theorems 3 and 4, we may need up to two times more coefficients to characterize the distribution). As a result, the number of coefficients used to represent the distribution may grow exponentially as we proceed with the rollup steps and conditioning steps.

To overcome the exponential growth of number of bases used in the representation, we develop an approximation algorithm to reorganize the Radon basis coefficients such that we can always keep compact representation of the distribution. This is possible since we used overcomplete Radon bases, which for any distribution, means there are more than one way to characterize it. Thus we can search for a sparse approximation representation for the distribution if it is not represented in a compact way.

5.3.1. Orthogonal Matching Pursuit. We note that after a series of mixing events happen, distributions on the homogeneous space X^k become smoother and their energies gradually concentrate to subspaces spanned by low-order Radon bases. More precisely, for any distribution $f \in L(X^k)$, if we consider the l_2 distance between f and the orthogonal projection of f to the subspace spanned by the columns in $\Phi^{j,k}$ $(0 \leq j \leq k)$, it is easy to prove that such a distance will decrease after each rollup step. For example, in the special case where $j = 0$, it reduces to the conclusion that the l_2 distance between f and the uniform distribution will decrease after each rollup step. In such sense, we should introduce lower-order Radon bases to see if we can more efficiently represent the distribution f after a series of mixing events.

ALGORITHM 3: Orthogonal Matching Pursuit Algorithm

Input: Basis dictionary Φ, distribution f, stopping criteria ϵ
Output: Residual r, coefficients x, indices Λ
Procedure:
while $\|r\| \geq \epsilon$ **do**
  Measure correlations $c \leftarrow \Phi^T r$
  $\Lambda \leftarrow \Lambda \cup \{\arg\max_j c(j)\}$
  $x \leftarrow \arg\min_{z:\text{supp}(z)\subset\Lambda} \|f - \Phi z\|_2$
  $r \leftarrow f - \Phi x$
end while

In Gilbert et al. [2003], a greedy algorithm, *Orthogonal Matching Pursuit (OMP)* (see Algorithm 3) is proposed to solve the sparse approximation problem over redundant dictionaries, which works by greedily searching for bases most correlated with the residual and using them to fit the distribution. The Orthogonal Matching Pursuit algorithm also has better theoretical guarantees about quality of the approximation, given that the dictionary has smaller incoherent parameter [Gilbert et al. 2003]. In our case, we can use the entire overcomplete Radon basis dictionary as input basis dictionary to OMP, yet with a large incoherent parameter. However, the computational burden of searching a combinatorial-size basis dictionary is unaffordable. We observe that in our probabilistic mixing model (which assumes that at each time-step only two targets may swap), targets will not get well mixed very quickly. For example, coefficients c_α on the set α will evenly spread their energy to other sets of the same size only if there is a target i in α which well mixes with all targets in α^c, with the basis $\Phi_{\alpha-\{i\}}$ being efficient in representing the distribution after these mixing events.

As a result, we may adaptively downsample the whole overcomplete Radon basis dictionary according to the current representation of f. Given $f = \sum_{\alpha\in I} \Phi_\alpha c_\alpha$, we downsample a subset of the basis dictionary, for example, $\{\beta : \beta \subset \alpha, |\alpha - \beta| \leq s, \alpha \in I\}$ (i.e., the basis β which is a subset of some $\alpha \in I$ and β differs from α at most s elements). For example, if $s = 1$ and $\alpha = \{1, 2, \ldots, k\} \in I$, then all $(k-1)$-sets of α together with α itself are sampled as candidate bases for approximation. If we use N bases to represent f, then we will downsample at most kN bases. With fewer bases sampled, we also achieve a smaller incoherence parameter.

For the computational complexity of the Orthogonal Matching Pursuit algorithm in our case, we note that, given $f = \sum_\alpha \Phi_\alpha c_\alpha$, the inner product of a basis Φ_β in the basis dictionary Φ and f can be computed in polynomial time because $\langle \Phi_\beta, f\rangle = \sum_\alpha \langle \Phi_\beta, \Phi_\alpha\rangle c_\alpha$, where

$$\langle \Phi_\beta, \Phi_\alpha\rangle = \begin{cases} \binom{n-|\alpha\cup\beta|}{k-|\alpha\cup\beta|} & \text{if } |\alpha\cup\beta| \leq k \\ \mathbf{0} & \text{if } |\alpha\cup\beta| > k \end{cases}.$$

Solving a least squares problem can also be done in polynomial time because the least squares solution is $(\Phi^T\Phi)^{-1}\Phi^T f$, where estimating $\Phi^T\Phi$ reduces to evaluating the inner product of two bases in the basis dictionary and estimating $\Phi^T f$ reduces to evaluating the inner product between a basis in the basis dictionary and f. So we have the following theorem.

THEOREM 5. *Suppose $f = \sum_\alpha \Phi_\alpha c_\alpha$, N is the number of nonzero coefficients c_α's, and k is the number of red targets. If we downsample at most kN bases and use the OMP algorithm to generate an m-term approximation solution, then the complexity of the OMP algorithm is $\mathcal{O}(k^2N(N+m)m + (k+m)m^3)$.*

Example 5.2. In the example in Figure 1(b), we see that the distribution f can be represented as

$$f = \Phi_{\{1,2\}} \cdot .5 + \Phi_{\{2,3\}} \cdot .25 + \Phi_{\{2,4\}} \cdot .25.$$

If we run OMP on f, the basis $\Phi^{\{2\}}$ and $\Phi^{\{1,2\}}$ can be identified which helps us to re-organize coefficients for f as

$$f \approx \Phi_{\{2\}} \cdot .25 + \Phi_{\{1,2\}} \cdot .25.$$

In such a simple example, the approximation is highly accurate.

5.3.2. Thresholding. We may also choose to do thresholding on Radon basis coefficients to maintain a sparse approximation to the distribution. Such a technique is especially useful after a sequence of observation events happen, since conditioning steps may result in exponential decay of Radon basis coefficients.

Given $f = \sum_\alpha \Phi_\alpha c_\alpha$, where c_α's are Radon basis coefficients. If we always insist on the positiveness of the Radon basis coefficients, then one can directly estimate the l_1 norm of $\Phi_\alpha c_\alpha$ as $\binom{n-|\alpha|}{k-|\alpha|} c_\alpha$. Based on l_1 norm contribution of $\Phi_\alpha c_\alpha$ to the distribution f, we can threshold off the insignificant bases to maintain a sparse approximation of f. Such an algorithm is quite scalable and in practice works well.

Finally we should note that OMP can help to represent the distribution on homogeneous spaces by lower-order bases while thresholding does not have such a property.

5.4. Distributed Approach

We assume that there is a leader-based tracking system in the sensor network, where a small number of leaders among the sensor nodes are responsible for computing and tracking the target properties. The main idea to achieve a distributed approach for rollup, conditioning, and sparse approximation is to decompose the space domain of the targets such that each leader maintains information about a set of targets that are close to it.

Before we describe the distributed approach, we assume the following.

(1) Each $1-$set $\{t_i\}$ $(1 \le i \le n)$ is always maintained in the system.
(2) We do not maintain the empty set and its coefficients. We notice from Section 4.1, it is known that $R(\Phi^{0,k}) \subset R(\Phi^{1,k})$, so we will not have information loss if all $1-$sets are maintained in the system.
(3) The normalizing step in the centralized conditioning step is not performed, as the distribution only differs by a multiplication constant if we do not do normalizing.

In the distributed sensor network, we have several leaders, for example, L_1, L_2, etc., keeping tracks of targets nearby. Figure 3 gives an example where the leader L_1 is surveiling tracks $\{t_1\}$, $\{t_2\}$, $\{t_3\}$ and $\{t_4\}$, and the leader L_2 is surveiling tracks $\{t_5\}$, $\{t_6\}$ and $\{t_7\}$ before the mixing and observation events happen. Each leader stores information about certain k-sets together with their Radon coefficients where the members of the k-sets are currently surveiled, for example, the leader L_1 has information about the track sets $\{t_1\}$, $\{t_3\}$ and $\{t_1, t_3\}$ with their Radon coefficients, and the leader L_2 has information about the track set $\{t_5\}$ and its Radon coefficient. Due to mixing and observation events previously occurred, it is also possible that, for example, the track set $\{t_3, t_5\}$ with its Radon coefficient are also incorporated to represent the underlying distribution. In such a case, we assume both leaders L_1 and L_2 store information about $\{t_3, t_5\}$ together with its Radon coefficient. More precisely, if a set with nonzero Radon coefficient has nonempty intersections with the set of tracks surveiled by different leaders, these leaders will store information about such a set at the same time. Thus, to ensure a set and its coefficient are maintained consistently among different leaders, these leaders

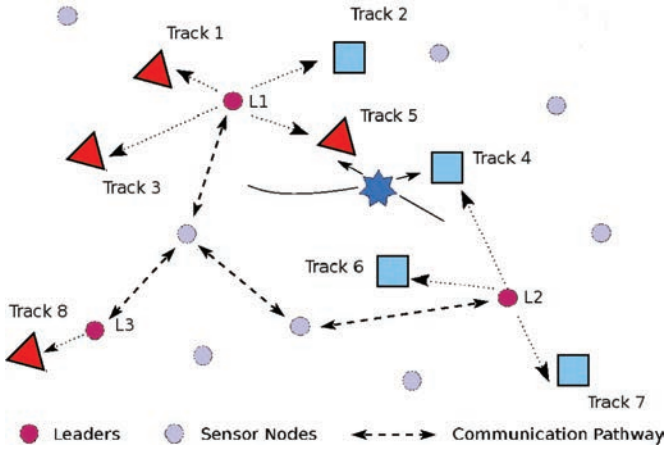


Fig. 3. An example scenario where targets on tracks t_4 and t_5 mix up and then separate. Then the target on track t_3 is revealed to be red.

communicate with each other when necessary. So we also assume each leader has its own communication pathways to other leaders in the network such that it can talk to them. As we will see later, such communication pathways will be updated accordingly with the updated observation events.

Leader nodes may change due to the movement of the targets, thus information maintained in one leader will be sent to the other leader when the leadership changes. Each leader needs to log the mixing and observation events that happened to the targets it surveils. We will see that part of the logs will be dropped when new observation events happen so that the total information stored within the leaders will be under control.

5.4.1. Rollup Step. At the rollup step where the targets on two tracks t_i and t_j are mixed, if the two targets are surveiled by the same leader, we simply perform the rollup step within the leader. If, however, the mixed two targets are surveiled by two different leaders, the two leaders, which presumably are physically close when the two tracks mix, will exchange their information of sets α's and coefficients c_α's currently maintained with each other to perform the rollup step. In both cases, the leaders gain all necessary information for the rollup algorithms. We will split the tracks surveiled by the two leaders according to the physical closeness information. The leaders which perform the rollup step operation also need to send part of the updated information to other leaders to ensure consistency.

Figure 3 depicts an example scenario, where 3 leaders L_1, L_2, and L_3 are in the network and surveil 8 tracks, the targets on tracks t_4 and t_5 mix up and then separate. Now the target on track t_5 is close to the leader L_1 and the target on track t_4 is close to the leader L_2. We will let the leader L_1 surveil tracks $\{t_1, t_2, t_3, t_5\}$ and let the leader L_2 surveil tracks $\{t_4, t_6, t_7\}$. In the example shown in Figure 3, if we assume before the rollup step, the leader L_1 maintains information about the sets $\{t_1\}$,$\{t_3\}$,$\{t_3, t_5\}$, and the leader L_2 maintains information about the set $\{t_5\}$. Then after the rollup step, the sets with nonzero coefficients are $\{t_1\}$, $\{t_3\}$, $\{t_3, t_5\}$, $\{t_4\}$, $\{t_5\}$ and $\{t_4, t_5\}$. Since we let the leader L_1 surveil tracks $\{t_1, t_2, t_3, t_5\}$, the sets $\{t_1\}$, $\{t_3\}$, $\{t_3, t_5\}$, $\{t_5\}$ and $\{t_4, t_5\}$ with their updated coefficients will be stored within leader L_1; we let the leader L_2 surveil tracks $\{t_4, t_6, t_7\}$, and the sets $\{t_4\}$ and $\{t_4, t_5\}$ with their updated coefficients will be stored in the leader L_2. We note that before the rollup step, the sets $\{t_3, t_5\}$ are maintained within both the leaders L_1 and L_2, but after the rollup step, only the set $\{t_4, t_5\}$ is maintained within both L_1 and L_2.

In particular, it might be possible that leader L_1 also maintains information about another set, for example, $\{t_5, t_8\}$ where the target on the track t_8 is currently tracked by another leader L_3. After the rollup step, a new set $\{t_4, t_8\}$ will emerge. The leader L_1 will then send the updated information about $\{t_5, t_8\}$ and $\{t_4, t_8\}$ to the leader L_3 via the communication pathway between them.

A pseudocode for the distributed algorithm for the rollup step is shown in Algorithm 4.

ALGORITHM 4: Distributed Algorithm for the Rollup Step

Input: Targets on tracks t_i and t_j mixed up and then separate.
Output: Updated sets and coefficients maintained in all leaders.
Procedure:
if t_i and t_j are surveiled by the same leader L_1 **then**
  Perform the rollup step as in algorithm 1 use data within L_1.
  for each set α maintained in L_1 **do**
    if α intersects with tracks surveiled by other leaders **then**
      L_1 send updated α and c_α to those leaders.
    end if
  end for
else if t_i and t_j are surveiled by different leaders L_1 and L_2 **then**
  Perform the rollup step as in algorithm 1 use data within L_1, L_2, split the updated information to the two leaders L_1 and L_2.
  for each set α maintained in L_1 **do**
    if α intersects with tracks surveiled by other leaders **then**
      L_1 send updated α and c_α to those leaders.
    end if
  end for
  for each set α maintained in L_2 **do**
    if α intersects with tracks surveiled by other leaders **then**
      L_2 send updated α and c_α to those leaders.
    end if
  end for
end if

5.4.2. Conditioning Step. At the conditioning step, when the property of a target on one track is revealed, we can perform the conditioning algorithm locally at the leader which surveils that track. In the centralized algorithm, whenever the target property on track t_j is revealed, we need to introduce all $\{t_j\} \cup \beta$ into the new representation for all sets β's currently maintained in the system. In the distributed algorithm, however, we slightly modify such a step. We incorporate $\{t_j\} \cup \beta$ where β contains the tracks that t_j has mixed with since the property of the target on the track t_j was lastly revealed.

For the example shown in Figure 3, we assume that the leader L_1 currently surveils tracks $\{t_1, t_2, t_3, t_5\}$ and it maintains information about the sets $\{t_1\}$, $\{t_3\}$, $\{t_3, t_5\}$, $\{t_5\}$, $\{t_4, t_5\}$ and $\{t_5, t_8\}$ where t_8 is another track currently surveiled by another leader L_3. If the property of the target on track t_3 is revealed to be red with high probability, then we can trace back along the track t_3 to see with which it mixes since the object on t_3 was revealed last time.

If for example, the track t_3 has mixed with tracks t_8, t_1, t_2, then we are going to incorporate $\{t_3\} \cup \beta$ where β satisfies the following.

(1) It is currently maintained in the leader L_1.
(2) It has nonempty intersection with $\{t_8, t_1, t_2\}$.

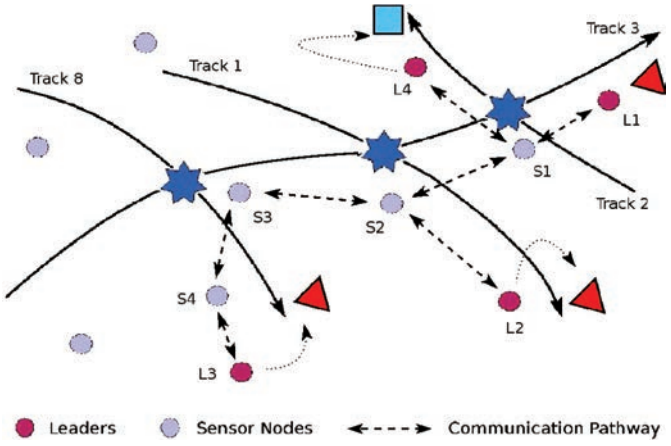


Fig. 4. Observation events happen and the communication pathways are accordingly updated.

Thus, new sets $\{t_1, t_3\}$, $\{t_3, t_5, t_8\}$ will be incorporated in the representation maintained by the leader L_1 (it is because $\{t_1\}$ and $\{t_5, t_8\}$ are the only two sets satisfying the two conditions). Besides, the coefficients on $\{t_1\}$, $\{t_5\}$, $\{t_4, t_5\}$, $\{t_5, t_8\}$ will be unchanged and the coefficients on $\{t_3\}$ and $\{t_3, t_5\}$ will be updated. Such a technique works efficiently in practice as the tracks mixed with t_3 recently are the most critical to address in the conditioning step.

When new sets $\{t_i\} \cup \beta$ are introduced, which may have nonzero intersections with tracks surveiled by other leaders, we need to send the updated information to these leaders. In the preceding case, only the information about $\{t_3, t_5, t_8\}$ needs to be sent to the leader L_3.

The communication pathways are consistently updated with the distributed conditioning step. To illustrate, see the example in Figure 4. In this example, when the target property of track t_3 is revealed, we can back trace along the track and identify that the sensor nodes S_1, S_2, and S_3 log the mixing events of t_3 with t_8, t_1, and t_2. We move forward from these sensors and identify pathways to the leaders that surveil the tracks t_8, t_1, and t_2, for example, from S_3 to S_4 to L_3, from S_2 to L_2, etc. Such a process naturally defines new communication pathways between leader L_1 and other related leaders. As a result, the communication pathways between these leaders are consistently updated with the process.

A pseudocode for the distributed algorithm for the conditioning step is shown in Algorithm 5.

5.4.3. Sparse Approximation Step. We have discussed two classes of sparse approximation algorithms in Section 5.3. The orthogonal matching pursuit algorithms can be performed within each leader independently whenever a leader has maintained too much information. However, if a set is maintained within different leaders, consistency of the Radon coefficients cannot be guaranteed if we run OMP in such a distributed way. Thus different leaders need to talk to each other to ensure consistency. Note that in practice, the sparsification operation will not happen very often. Moreover, only information about sets that have nonempty intersections with tracks surveiled by other leaders need to be sent to other leaders. So the total amount of information to be sent is still under control. The thresholding techniques are fully distributive, that is, whenever a k-set has coefficient less than a threshold, it would be thrown away within each leader accordingly. In this case, each leader can perform the operation of thresholding independently, and the result will always be consistent.

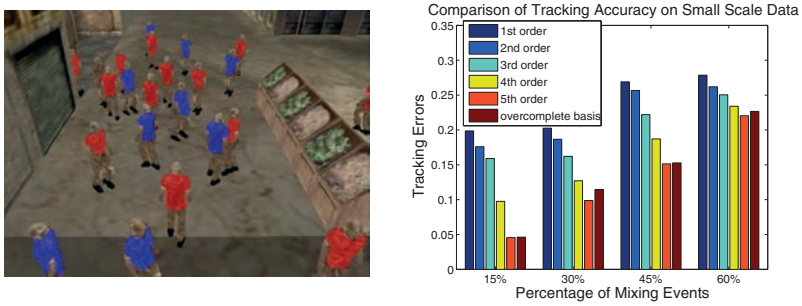
ALGORITHM 5: Distributed Algorithm for the Conditioning Step

Input: Targets on tracks t_i is revealed to be red with high probability.
Output: Updated sets and coefficients maintained in all leaders.
Procedure:
Identify the mixing events of t_i since t_i is revealed last time.
Perform the conditioning step as in algorithm 2 use data within L_1, taking account of the recent mixing events of t_i.
for each set α maintained in L_1 whose value c_α changed **do**
  if α intersects with tracks surveiled by other leaders **then**
    L_1 send updated α and c_α to those leaders.
  end if
end for
for each track t_j that t_i mixed with recently **do**
  Identify the sensor node which logged the mixing event.
  Identify the leader currently surveils t_j.
  Update the pathway between leaders that surveil t_i and t_j.
end for

5.5. Discussion

In this section, we give brief discussion on several issues related to the proposed algorithms.

—*Error Propagation.* Whenever we do sparse approximation to approximate f by f', we introduce errors. Mixing events always shrink the approximation error $\|f - f'\|$ which is due to the fact that mixing matrices have eigenvalues bounded by 1, while observation events do not. However, we note that n independent observations of all the targets will drive both f and f' to converge to delta distributions. In this sense, approximation errors can be under control.
—*Sparsity Propagation.* Theoretically one may need up to two times more coefficients to represent the updated distribution. In practice, if mixing and observation events happen locally, sparsity can always be kept to a relatively low level.
—*Timing for Sparsification.* In practice, we do sparsification after a sequence of mixing events happen or a sequence of observations happen such that the true distribution becomes more smooth or peaky while the coefficients used are too many.
—*Distributed Algorithm.* Compared with the centralized algorithm, the distributed approach divides the storage of information into different leaders such that processing information for rollup, conditioning, and sparse approximation step can be mostly performed locally. In such a sense, we can still save a great amount of computational time though communication overhead is incurred.
—*Positiveness of Coefficients.* Clearly, rollup and conditioning algorithms will keep the positiveness of the coefficients. However, the OMP algorithm may result in approximating the distribution with negative coefficients. In practice, we preserve positiveness of the coefficients by projecting on the positive cone.
—*Prediction.* By using pseudoinverse $(\Phi^{j,k})^+$ mentioned in Section 4.1, we are able to compute the score for each target. We pick out k tracks which have the highest scores and predict the targets on these k tracks are red. This is equivalent to looking at orthogonal projection of f to subspaces spanned by $\Phi^{1,k}$ and finding out which k-set has the largest weight.
—*Comparative Methods.* Similar as the approaches taken in Huang et al. [2009a] in studying the identity management problem, we can approximate the distribution over X^k by using low-order Radon bases, that is, $f \approx R^{j,k} c_j$ with small j's so that we only store c_j which is a vector of length $\binom{n}{j}$. Though polynomial in n, low-order Radon



(a) (b)

Fig. 5. (a) A view of the simulated data. (b) Tracking errors with different percentages of mixing events.

bases are incapable of characterizing peaky distribution over X^k. In Section 6, we will compare our approach with using only low-order Radon basis coefficients.

—*k Is Not Known as A Priori.* For a more general case where k - the number of red targets is not known as a priori, we can still use the Radon bases to address the problem. Essentially the rollup step and the conditioning step will be unaffected, because we can see that the algorithm for the rollup step is independent of k, while the algorithm for the conditioning step can also be adapted to the case that k is not known (we just pretend the "if" statement is not there). For the sparse approximation step, we note that since the spaces spanned by Radon bases of different orders have a particular hierarchical structure, thus we can approximate high (low)-order Radon bases coefficients with low (high) nearing-order Radon bases coefficients, depending on whether the scenario is highly uncertain (certain). Such an operation will be independent of k.

—*Connection between Property Management and Identity Management.* As for the difference between property management and identity management, property management tries to infer less information and therefore can be easier and less costly to implement. At the same time, in many settings, information about properties can be sufficient for the network needs, for example, to differentiate friend from enemy. In general, the approach for the property management problem provides an alternative method for the identity management problem. From a mathematical point of view, suppose we have n targets, then we can code them using $\mathcal{O}(\log n)$ bits such that each identity has a unique binary code of length $\mathcal{O}(\log n)$. For each bit, all the targets can be classified as either red or blue depending whether the bit is 0 or 1. Then, based on probabilistic beliefs on each bit, we can infer the target identity. One can also generalize such an approach where properties of the targets act as features which collaboratively determine the target identities. On the other hand, it turns out that the Fourier basis coefficients used for the identity management problem [Huang et al. 2009a, 2009b] can be collapsed to the Radon basis coefficients discussed in this article. The intuition is that if we have a distribution over the permutation group (each permutation assigns target positions to the identities), and if the first k identities have color property red, then by summing up the probabilities over all permutations such that it maps a particular k-set to the first k identities, that is, we don't care about the permutation within, essentially we are collapsing the Fourier basis coefficients for permutation groups to the Radon basis coefficients for the homogeneous spaces. Thus, using Radon bases to address the property management problem can be viewed as an approach which collapses the probability distributions given by the Fourier coefficients for the identity management problem.

6. EVALUATION

In this section, we perform several experiments to illustrate the effectiveness and efficiency of the proposed approach. We use the Delta3D game engine to generate simulated crowds of up to 100 moving targets wearing either red or blue clothes and walking around in an outdoor market [Heath and Guibas 2008]; Figure 5(a) depicts a snapshot view of the simulated crowd. Such a simulation approach allows us to obtain a more accurate ground truth for big crowds than was feasible in the usual physical testbed. The data contains interesting movement patterns and we can extract mixing and observation events directly from the data. We log a mixing event whenever two targets get close to each other within some distance and an observation event whenever one target is separated from all the other targets for some distance. We can control the percentages of mixing events by adjusting the distance parameters as well. We measure tracking errors using the fraction of mislabeled target properties over the tracks.

We first run a small-scale experiment where there are 10 targets, 5 red and 5 blue. The homogeneous space X^5 is of size $\binom{10}{5} = 252$. As illustrated in Figure 5(b), four sets experiments with different percentages of mixing events were performed, reflecting scenarios of high certainty to high confusion. For each set of experiments, we run the centralized algorithm and compare with using only low-order Radon basis coefficients. We measure tracking errors using the fraction of mislabeled target properties over the tracks. When mixing events happen rarely, using high-order Radon basis coefficients can greatly help to improve tracking accuracy; while if mixing events happen frequently, using high-order Radon basis coefficients does not help much to improve tracking accuracy. This is reasonable since if mixing events happen rarely, distributions can be well-characterized by a high-order Radon basis while low order bases are not sufficient to characterize distributions; on the other hand, if mixing events happen frequently, distributions can be well-characterized by low-order Radon bases, so using a high-order Radon basis would not provide additional benefits. Our overcomplete basis approach uses on average about 50 bases to characterize the distribution over X^5. The tracking accuracy is almost comparable to completely storing the distribution on X^5, which requires storing 252 coefficients.

In the small-scale experiments, the approach that uses 5-th-order Radon bases can be regarded as an optimal approach in terms of tracking accuracy. It's the best that we can do, which completely keeps track of the distributions over X^5. The approach that uses 1st-order Radon bases can be viewed as a baseline algorithm. In such an approach, we use 10 numbers to indicate the likelihoods of targets being red or blue, while ignoring the mutual dependence structure among the targets.

From the small-scale experiment, we can see that there is a fundamental trade-off between the number of coefficients used and the tracking accuracy. With more bases used, we can track targets better, however, we cannot use as many bases as we want if the problem size goes large because in the extreme case we would use exponentially many bases. Moreover, the Heisenberg uncertainty principle plays an important role in our experiments. The scenario where there are very few mixing events can be well-characterized by using only the low-order Radon bases; while the scenario where there are a lot of observation events can be well-characterized by using only the high-order Radon bases. That's why we see greater improvements of the tracking accuracy by using more high-order Radon bases for the case where the targets have few mixings; while there are relatively smaller improvements of tracking accuracy by using more high-order Radon bases if targets are well-mixed. Thus, using an overcomplete Radon basis is a good way to balance the high tracking accuracy requirement and computational efficiency.

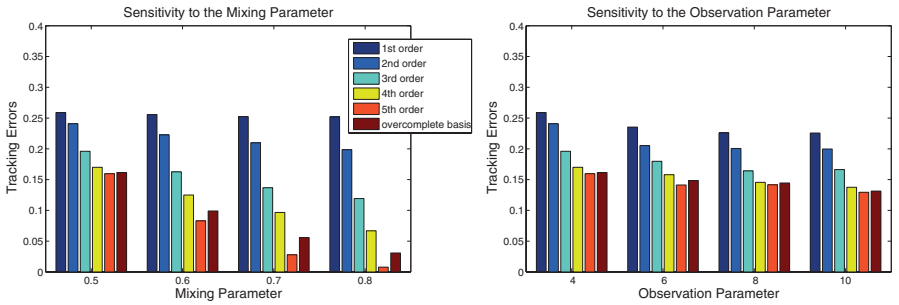


Fig. 6. Sensitivity of tracking accuracies with respect to the mixing and observation parameters.

Table III. Large-Scale Experiments

Methods	#Targets	Tracking Errors	Running Time
1st Order Radon Basis	$n = 20$	0.3167	0.02(s)
	$n = 60$	0.2804	0.17(s)
	$n = 100$	0.2836	0.46(s)
3rd Order Radon Basis	$n = 20$	0.2977	29.79(s)
	$n = 60$	0.2845	119.45(s)
	$n = 100$	0.2891	912.21(s)
Overcomplete Basis	$n = 20$	0.1727	29.69(s)
	$n = 60$	0.1751	292.90(s)
	$n = 100$	0.1823	1342.35(s)

In our experiments, we can control two sets of parameters which determine the tracking quality: one is the swapping probability, that is, if we can keep track of who is who when two targets mix with high probability during the rollup step, we can achieve better tracking performance; the other is the likelihood function, that is, if the likelihood for observing the property of a target is high, then a conditioning step can resolve the ambiguities better. We explore the sensitivity of tracking accuracy with respect to different swapping probability and likelihood function parameters. As depicted in Figure 6, as the mixing parameter goes larger, or the likelihood parameter goes larger, we can get better tracking accuracy. In general, the tracking accuracy is more sensitive with respect to the mixing parameter. This means that if we can robustly estimate who is who when mixing events happen, we can greatly improve the tracking results. Nevertheless, in all of the cases, using an overcomplete Radon bases dictionary can yield us very good results.

Our algorithm shows great benefits on tracking accuracy and computational time in larger-scale experiments. When there are n ($n \geq 20$) targets and half of them are red, it would be impossible to store the entire distribution f, which means that we cannot realize the optimal approach. Thus we compare our approach with using only low-order Radon basis coefficients which approximates f, that is, we approximate f by $R^{j,k}c_j$ with small j. In our experiments, there are 1000 time steps and half of them are mixing events. We use up to hundreds of bases in the centralized overcomplete basis algorithm to characterize the distribution. From Table III, we see that our approach improves the tracking accuracy greatly compared with only using low-order Radon basis coefficients and the running time of our approach is comparable to using Radon basis coefficients of order 3. We also run the large-scale experiments to see how the tracking accuracy varies with the percentages of mixing events. As shown in Figure 7(a), the tracking errors increase when mixing events happen more frequently, which coincides with the intuition. The tracking errors also increase with the increment of the number of

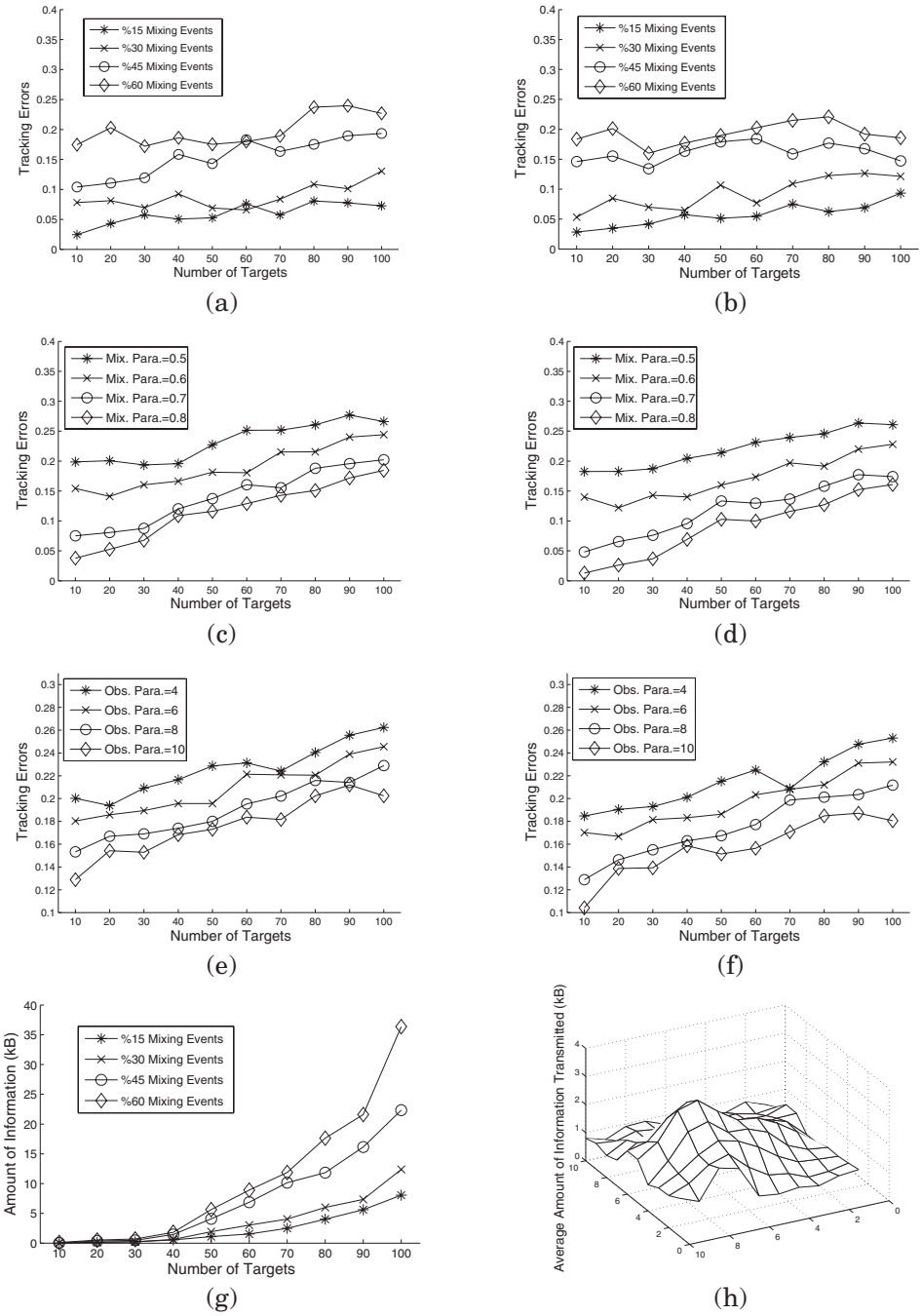


Fig. 7. (a) Tracking errors with large-scale data. (b) Tracking errors with the distributed approach. (c) Processing overhead at the leaders with distributed approach. (d) Traffic distribution over the sensor network.

Table IV. Identity Management Tracking Accuracy

Methods	Fourier Approach (order)	Radon Approach (#Properties)
$n = 20$	0.3842 (1)	0.4543 (5)
	0.3740 (2)	0.4189 (6)
	0.3603 (3)	0.3592 (7)
$n = 60$	0.3811 (1)	0.4428 (6)
	0.3701 (2)	0.3741 (7)
	0.3595 (3)	0.3624 (8)
$n = 100$	0.3814 (1)	0.4250 (7)
	0.3569 (2)	0.3809 (8)
	0.3499 (3)	0.3447 (9)

targets. The explanation for this is that we use a linear growing number of bases to approximate a distribution whose size is exponentially growing.

We also simulate a distributed sensor network environment and test the performance of the proposed distributed approach. We simulate a scenario where 100 sensors are deployed approximately on a regular 10 by 10 grid. We compute the tracking accuracy with different number of targets and different percentage of mixing events. Here 50–500 Radon basis coefficients are used to approximate the distributions for the scenarios where there are 10–100 targets (half of them are red) moving within the field, which is comparable to the number of basis coefficients we used to test the centralized algorithm. From Figure 7(b), we see that the tracking errors are comparable to the centralized algorithm depicted in Figure 7(a). In practical sensor network applications, the total amount of information stored in the distributed sensor network can be much larger than the centralized case. We expect the tracking accuracy can be further reduced if we incorporate more bases to represent the distributions.

We also perform sensitivity analysis of the tracking accuracy with respect to different swapping probability and likelihood function parameters, in much the same way as we did for the small-scale experiments. As depicted in Figure 7(c)–(f), we will have better tracking accuracy as the mixing parameter go larger, or likelihood parameters go larger, both of which correspond to cases where there are more certainties. The distributed approach tends to have a comparable tracking accuracy to the centralized approach under various sets of parameters.

To evaluate the overhead of the distributed approach in practical sensor networks, we also test the average amount of information processed at the leaders and the amount of traffic transmitted within the network. The experiment is performed with 100 targets moving in the network. Figure 7(g) depicts the average amount of information processed at the leader nodes. We can find that the leaders need to process more information as the number of targets grows. When observation events happen less frequently, the amount of information processed within the leaders will be smaller, which is due to the fact that more leaders are involved in tracking. Figure 7(h) depicts the amount of traffic transmitted over the sensor network. We see that each sensor transmits no more than 4kB information to other nodes which is not a heavy load with sensor network settings. Sensors in some part of the network where mixing and observation events happen frequently have larger amount of information transmission. The overall traffic distribution, however, is smooth over the entire sensor network.

We finally set up an experiment which compares using the properties to collaboratively determine the target identities with the existing Fourier approach for the identity management problem. In this experiment, each target has many properties which collaboratively determine its identity. It can be seen from Table IV that the more features are used, the better tracking accuracy can be achieved. The tracking accuracy of two approaches are comparable.

7. CONCLUSION

In this article, we have studied the property management problem. A novel method which uses an overcomplete Radon basis dictionary to represent uncertainties is proposed. We developed scalable algorithms to efficiently update the Radon basis coefficients, together with approximation algorithms which can maintain a sparse approximation of the true distribution. Based on this, we further proposed a distributed approach which can be practically implemented for a distributed sensor network. Compared with other possible methods, the proposed approach achieves better performance in tracking accuracy with tolerable computation and communication overhead.

REFERENCES

ARORA, A., DUTTA, P., BAPAT, S., KULATHUMANI, V., ZHANG, H., NAIK, V., MITTAL, V., CAO, H., GOUDA, M., CHOI, Y., HERMAN, T., KULKARNI, S., ARUMUGAM, U., NESTERENKO, M., VORA, A., AND MIYASHITA, M. 2004. A line in the sand: A wireless sensor network for target detection, classification, and tracking. *Comput. Netw. 46*, 605–634.

ASLAM, J., BUTLER, Z., CONSTANTIN, F., CRESPI, V., CYBENKO, G., AND RUS, D. 2003. Tracking a moving object with a binary sensor network. In *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys'03)*.

BAR-SHALOM, Y. AND FORTMANN, T. E. 1988. *Tracking and Data Association*. Academic Press.

COLLINS, J. AND UHLMANN, J. 1992. Efficient gating in data association with multivariate distributed states. *IEEE Trans. Aerospace Electron. Syst. 28*, 909–916.

COX, I. AND HINGORANI, S. 1994. An efficient implementation of reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. In *Proceedings of the 12th International Conference on Pattern Recognition, Vol.1-Conference A: Computer Vision and Image Processing.*

DIACONIS, P. 1988. *Group Representations in Probability and Statistics*. Institute of Mathematical Statistics.

DUTTA, P., HUI, J., JEONG, J., KIM, S., SHARP, C., TANEJA, J., TOLLE, G., WHITEHOUSE, K., AND CULLER, D. 2006. Trio: Enabling sustainable and scalable outdoor wireless sensor network deployments. In *Proceedings of the 5th International Conference on Information Processing in Sensor Networks (IPSN'06)*.

GILBERT, A., MUTHUKRISHNAN, S., AND STRAUSS, M. J. 2003. Approximation of functions over redundant dictionaries using coherence. In *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'03)*.

GUIBAS, L. J. 2008. The identity management problem – A short survey. In *Proceedings of International Conference on Information Fusion*.

HEATH, K. AND GUIBAS, L. 2008. Multi-person tracking from sparse 3d trajectories in a camera sensor network. In *Proceedings of the 2nd ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC'08)*.

HUANG, J., GUESTRIN, C., AND GUIBAS, L. J. 2009a. Fourier theoretic probabilistic inference over permutations. *J. Mach. Learn. Res. 10*, 997–1070.

HUANG, J., GUESTRIN, C., JIANG, X., AND GUIBAS, L. J. 2009b. Exploiting probabilistic independence for permutations. In *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS'09).*

JIANG, X., YAO, Y., AND GUIBAS, L. 2009. Stable identification of cliques with restricted sensing. In *Neural Information Processing Systems (NIPS) - Learning with Orderings Workshop*.

KONDOR, R., HOWARD, A., AND JEBARA, T. 2007. Multi-object tracking with representations of the symmetric group. In *Proceedings of the 11th International Conference on Artificial Intelligence and Statistics (AISTATS'07).*

LIU, Y., ZHU, Y., NI, L. M., AND XUE, G. 2011. A reliability-oriented transmission service in wireless sensor networks. *IEEE Trans. Parallel Distrib. Syst. 22,* 12, 2100–2107.

MO, L., HE, Y., LIU, Y., ZHAO, J., TANG, S.-J., LI, X., AND DAI, G. 2009. Canopy closure estimates with greenorbs: Sustainable sensing in the forest. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys'09)*.

MURTY, K. 1968. An algorithm for ranking all the assignments in order of increasing cost. *Oper. Res. 16*, 682–687.

OH, S., RUSSELL, S., AND SASTRY, S. 2004. Markov chain monte carlo data association for general multipletarget tracking problems. In *Proceedings of the 43rd IEEE Conference on Decision and Control (CDC'04)*.

OH, S. AND SASTRY, S. 2005. A polynomial-time approximation algorithm for joint probabilistic data association. In *Proceedings of the American Control Conference*.

POORE, A. 1995. Multidimensional assignment and multitarget tracking. In *Proceedings of the Partitioning Data Sets (DIMACS Workshop)*. Vol. 19, 169–196.

REID, D. 1979. An algorithm for tracking multiple targets. *IEEE Trans. Autom. Control 6*, 843–854.

SAGAN, B. 2001. *The Symmetric Group: Representations, Combinatorial Algorithms, and Symmetric Functions*. Springer.

SCHUMITSCH, B., THRUN, S., BRADSKI, G., AND OLUKOTUN, K. 2005. The information-form data association filter. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS'05)*.

SERRE, J.-P. 1977. *Linear Representation of Finite Groups*. Springer.

SHIN, J., GUIBAS, L. J., AND ZHAO, F. 2003. A distributed algorithm for managing multi-target identities in wireless ad-hoc sensor networks. In *Proceedings of the 2nd International Conference on Information Processing in Sensor Networks (IPSN'03)*.

XIE, D., YAN, T., GANESAN, D., AND HANSON, A. 2008. Design and implementation of a dual-camera wireless sensor network for object retrieval. In *Proceedings of the 7th International Conference on Information Processing in Sensor Networks (IPSN'08)*.

YANG, Z., LIU, Y., AND LI, X. 2010. Beyond trilateration: On the localizability of wireless ad-hoc networks. *IEEE/ACM Trans. Netw. 18,* 6, 1806–1814.

ZHU, Y., LIU, Y., AND NI, L. M. 2012. Optimizing event detection in low duty-cycled sensor networks. *Wirel. Netw. 18,* 3, 241–255.

Received April 2011; revised April 2012; accepted April 2012