Received October 30, 2015, accepted November 16, 2015, date of publication December 24, 2015,
date of current version January 7, 2016.

Digital Object Identifier 10.1109/ACCESS.2015.2512380

Tensor Voting Techniques and Applications in Mobile Trace Inference

ERTE PAN, (Student Member, IEEE), MIAO PAN, (Member, IEEE), AND ZHU HAN, (Fellow, IEEE)
Department of Electrical and Computer Engineering, University of Houston, Houston, TX 77004, USA

Corresponding author: Z. Han (zhan2@uh.edu)

The work of M. Pan was partially supported by the U.S. National Science Foundation under Grant CNS-1350230 and Grant CNS-1343361.

ABSTRACT Initially appearing as an abstract object frequently used in math and physics, tensors have been attracting increasing interest in a broad range of research fields, such as engineering and data science. However, a few studies have addressed their application in wireless scenarios. In this paper, we investigate the wide applications of tensor techniques with an emphasis on the tensor voting method, which serves as an artificial intelligence approach for automatic inference and perceptual grouping. To illustrate the efficiency of the tensor voting approach, we tackle the tracking problem of inferring human mobility traces, which can provide key location information of networking objects. The trace inferring problem is considered under the circumstance that the recorded location information exhibits missing data and noise. Based on the tensor voting theory, we propose a sparse tensor voting algorithm and an implementation scheme with computational efficiency. The model is constructed based on the geometric connections between the input signals and encodes the structure information in the tensor matrix. The missing location information and noise can be distinguished via tensor decomposition. Once the trace information has been completed, further analysis of the inferred trace can be performed based on feature extraction to differentiate different objects. Moreover, we propose several feature extraction methods to characterize the inferred trace, including the scale invariant feature obtained from the fractal analysis. The proposed methods for trace completion and pattern analysis are applied to real human mobility traces. The results show that our proposed approach effectively recovers human mobility trace from the incomplete and noisy data input, and discovers meaningful patterns of inferred traces from various objects.

INDEX TERMS Motion tracking, trace inference, normal space, sparse tensor voting, trace analysis, fractal dimension, Fourier descriptor.

I. INTRODUCTION

Tensor theories nowadays have been widely applied to engineering problems and big data applications. From the numerical point of view, tensors are the extension concept of scalar, vector and matrix. In addition to the mathematical formation, tensors can be employed with specific physical meanings under different contexts. For instance, a tensor can be used to represent a geometric object whose geometric property is invariant to the coordinate systems. Also, a tensor can be utilized to describe linear relation between vectors, scalars and other multidimensional arrays. The relation can be mathematically expressed as a multi-linear mapping. As a powerful and popular tool, tensor theories have been employed in various research and pragmatic fields. In mechanics, the stresses are presented by tensors, which brings about concise modelization and efficient computation [1]. In data science, tensors are utilized to model the data cube in which the inherent property of data is encoded and can be revealed through tensor decomposition [2]. In image processing [3], tensors can be used to model the geometric objects such as the normal space, which contributes to the inference of grouping points. As one of tensor theories, tensor voting [4] is the artificial intelligence technique widely used for automatic perceptual grouping where the tensor voting algorithm estimates and infers geometric objects based on the principles derived from human psychology. In this paper, we focus on the tensor voting algorithm and study the tracking problem as an illustration for the sake of rapid development of networking, machine learning and signal processing which bring up many tracking issues of constant interests.

Significant research efforts nowadays have been devoted to the challenges and problems in mobile networks as a result of fast growth of wireless networks [5], [6]. As the innovative development of personal devices such as smart phones, a prominent amount of location services are emerging to serve individual user for various purposes.

For example, LTE-Direct and device-to-device require location information so as to enable discovering nearby devices and their services. However, the location information might be missing for indoor applications due to loss of signals or incomplete even for the outdoor GPS data. It is therefore important and challenging to retrieve the complete device tracking information to realize various location-based functions of wireless networks. Intensive research efforts have been focused on the tracking problem from diverse aspects, ranging from mobility trace study to the image processing field. Work in [7] investigates the patterns of human walk traces from the statistical point of view. The synthetic features of human walk trace are captured by the proposed mobility model. Rhee *et al.* [8] employ the random walks model called Levy-walk, to emulate the characteristics reflected in the walking patterns. It proves that the statistical similarity does exist between the random walk model and human walks.

As an example to illustrate tensor methods, we address the tracking problem of inferring human mobility trace under the circumstance that the recorded location information exhibits missing and noisy data. Trace inference provides key location information of objects such as personal devices in wireless networks, and can serve as one of the important networking topics. Based on the tensor voting theory, we propose an efficient tensor voting algorithm and a specified implementation scheme. The model is constructed based on the geometric connections between the input signals and encodes the structure information in the tensor matrix. Thus, the computation is carried out in the form of matrix, which reduces the computation load. The proposed method is applied to real human mobility trace. After trace completion by tensor voting, we develop a systematic framework of feature extraction to analyze the traces. The fractal analysis is incorporated in the feature extraction method to characterize scale-independent features of human mobile trace. The results show that our proposed approach effectively recovers human mobility trace from the incomplete data input and provides comprehensive analysis of the trace. Our key contributions are:

1) Giving the detailed derivation of constructing the normal space based on the eigenvalue problem;
2) Implementing the tensor voting algorithm efficiently in the sparse sense;
3) Applying valid evaluation criteria to quantify the performance of the proposed tensor voting algorithm;
4) Applying fractal analysis to characterize the trace features for data mining tasks.

This paper is based on our previous work [9] and organized as follows: In Section II, the literature survey is presented to provide a broad view of tensor methods as well as the fractal analysis utilized in this paper. In Section III, the mathematical model of tensor voting is introduced to encode the data and perform the structure inferring procedure. The inference algorithm is given in Section IV. In Section V, trace analysis is performed based on the efficient feature extraction including fractal analysis. Simulation results are presented in Section VI. In Section VII, we draw conclusions and discuss potential future work.

II. LITERATURE SURVEY

Some tracking problems can be essentially transformed as the image processing problems where the tensor voting technique has been widely utilized. Guy elaborated tensor voting theory with insightful analysis in his Ph.D. thesis [10]. The theory is then ameliorated by other researchers in the past decade [11]. In tracking applications, as the topic of this paper, tensor voting systematically infers hidden or incomplete structures, for instance, gaps and broken parts in the trace curve. Tensor voting is also referred to as perceptual grouping [12] emphasize the contribution of the Gestalt principles on which the theory is based. In brief, the Gestalt principles state that the presence of each input token (site, pixel, signal, etc.) implies a hypothesis that the structure passes through it. For example, considering the $2D$ imaging process of a chair, if one pixel has recorded the chair signal, it is highly likely that some of its neighboring pixels should have captured the same structure/object signals. In other words, it is human nature to configure simple elements into the perception of complex structures. In [13] and [14], the object signals observed by fixed cameras are represented using spatiotemporal features which facilitates the application of tensor voting theory. The advantage of applying tensor voting brings several geometric properties including smooth continuous trajectories and bounding boxes with minimum registration error. Although there are extensive research efforts dedicated to tensor voting study in the image processing field [15], little work involving the tensor voting theory has been done in the communication realm according to the best of our knowledge. Moreover, very few research handles with the missing data problem in the tracking context.

Besides tensor voting, tensor theories have been widely applied to emerging hot topics such as big data. Tensor decomposition has been developed to address various data mining tasks as an extension of principal component analysis (PCA) in higher dimensions. In [16], a scalable and distributed version of the Tucker model, MR-T, is implemented using the Hadoop MapReduce framework. Liavas and Sidiropoulos [17] propose a new constrained tensor factorization framework, building upon the Alternating Direction method of Multipliers (ADMoM). Work in [18] permeates benefits from rank minimization to scalable imputation of missing data, via tracking low-dimensional subspaces and unraveling latent structure from incomplete streaming data. Work in [19] addresses the problems of computing decompositions of full tensors by using compressed sensing (CS) methods working on incomplete tensors, i.e., tensors with only a few known elements.

Another topic in tracking problems is trajectory analysis [20], which has wide application scenarios including vehicle traffic management, vessel classification by satellites images and so forth. In [21] a unifying framework is constructed to mine trajectory patterns of various

temporal tightness. The proposed framework consists of two phases: initial pattern discovery and granularity adjustment. Mouillot and Viale [22] employ the fractal analysis of a fin whale's trajectory tracked by the satellite. The implemented fractal analysis provides the scale-independent measurement to summarize interactions between an organism and its ecosystem and depends on the heterogeneity of the whale's environment and the whale's ability to perceive it. In [23], several real-world human mobility traces are employed to analyze network robustness in the time domain. Liu and Li [24] propose a novel integrated framework for multiple human trajectory detection, learning and analysis in complicated environments. In [25], a new approach for abnormal loitering detection using trajectory analysis is described and Inverse Perspective Mapping (IPM) is presented to resolve distortion of trajectory direction.

III. TENSOR VOTING MODEL

In this section, we illustrate the tensor voting framework. In Subsection III-A, we present the way to encode the normal space with tensor representation. In Subsection III-B, the fundamental stick tensor voting is addressed as the basis for inferring geometric structure via the encoded normal space. In Subsection III-C, the initialization procedure for tensor voting is introduced under the circumstance that no prior structure information is known. In Subsection III-D, the inference method based on tensor decomposition is explained.

In order to infer the hidden or missing structures, we need to model the structures mathematically first. Generally, the structure types in the form of $2D$ images can be classified into two categories: curves and regions. Curves are modeled as the structures that have a 1-d normal space, which is referred to as a stick. Regions are modeled as the structures that have a 2-d normal space, referred to as a ball. Normal space represents the structure types well, but it is required to know how salient the structures are in order to adequately model the structure. Hence, the parameter defined as saliency associated with each structure type is employed to indicate the size of structure. Both the normal space and saliency information are encoded in a tensor via specific calculation that will be described later.

After developing the mathematical models, we can further explain the hints obtained from the Gestalt principles [26]: (1) a token "communicates" its structure information to its surrounding tokens in a certain way with respect to its normal space, i.e., the surrounding tokens under its influence are supposed to have the same kind of normal space; (2) in the real world, a token may contain a combination of information of both structure types. For instance, a token that actually belongs to a curve has a dominant saliency in the 1-d normal space while it probably has minor saliency in the 2-d normal space.

A. ENCODE NORMAL SPACE WITH TENSOR

In math, a normal space is an N-by-N matrix for objects in N-dimension, denoted by N_d. Consider a d-dimension normal space in the N-dimension world, which is spanned by the first d out of N orthonormal basis vectors $\mathbf{e}_k$, $k = 1, 2, \ldots, N$, we have the normal space matrix expressed as:

$$\mathbf{N}_d = \sum_{k=1}^{d} \mathbf{e}_k \mathbf{e}_k^T. \tag{1}$$

The projection of any vector $\mathbf{v}$ into this normal space is:

$$\mathbf{v}_n = \mathbf{N}_d \mathbf{v}. \tag{2}$$

It can be easily proved that $\mathbf{v}_n$ is the projection of $\mathbf{v}$ into the normal space by showing that: (1) $\mathbf{v}_n$ can be linearly expressed by the basis vectors $\mathbf{e}_k$, $k = 1, 2, \ldots, N$; (2) dot product $\langle \mathbf{v}_n - \mathbf{v}, \mathbf{v}_n \rangle = 0$.

The exact tensor matrix $\mathbf{V}$ that encodes both normal spaces and their saliency is assumed to be known. From the previous discussion, we know that $\mathbf{V}$ is a symmetric, positive semi-definite N-by-N matrix. Suppose its eigenvalues are ordered as $\lambda_1 \geq \ldots \geq \lambda_N \geq 0$ with corresponding eigenvectors $\mathbf{e}_1, \ldots, \mathbf{e}_N$. By the knowledge of linear algebra, we know that the eigenvectors of $\mathbf{V}$ are orthogonal with each other (for the eigenvectors that belong to 0 eigenvalue, we can generate those eigenvectors in a way that they meet this requirement). Then if we normalize these eigenvectors into unit vectors, we have a set of orthonormal basis $\hat{\mathbf{e}}_1, \ldots, \hat{\mathbf{e}}_N$. Furthermore, we have the following derivations,

$$\mathbf{V}\hat{\mathbf{e}}_d = \lambda_d \hat{\mathbf{e}}_d, \tag{3}$$

$$\mathbf{V}\hat{\mathbf{e}}_d \hat{\mathbf{e}}_d^T = \lambda_d \hat{\mathbf{e}}_d \hat{\mathbf{e}}_d^T, \tag{4}$$

$$\mathbf{V} \sum_{d=1}^{N} \hat{\mathbf{e}}_d \hat{\mathbf{e}}_d^T = \sum_{d=1}^{N} \lambda_d \hat{\mathbf{e}}_d \hat{\mathbf{e}}_d^T, \tag{5}$$

$$\mathbf{V} = \sum_{d=1}^{N} \lambda_d \hat{\mathbf{e}}_d \hat{\mathbf{e}}_d^T = \sum_{d=1}^{N-1} (\lambda_d - \lambda_{d+1}) \sum_{k=1}^{d} \hat{\mathbf{e}}_k \hat{\mathbf{e}}_k^T + \lambda_N \sum_{k=1}^{N} \hat{\mathbf{e}}_k \hat{\mathbf{e}}_k^T, \tag{6}$$

$$\mathbf{V} = \sum_{d=1}^{N-1} (\lambda_d - \lambda_{d+1}) N_d + \lambda_N N_N. \tag{7}$$

From (7), we define the saliency s_d in the straight forward fashion: $s_d = \lambda_d - \lambda_{d+1}$, if $d < N$; $s_d = \lambda_N$, if $d = N$. Thus, substituting the saliency into (7), we have,

$$\mathbf{V} = \sum_{d=1}^{N} s_d N_d. \tag{8}$$

B. INFERRING STRUCTURE

To simplify the illustration procedure, we will assume that we already know the structure types and saliencies for now. We will discuss how to obtain this information later.

We start with the simplest case that is the fundamental unit stick vote when the normal space is 1-d. Consider a voter point p (a token that passes its structure information

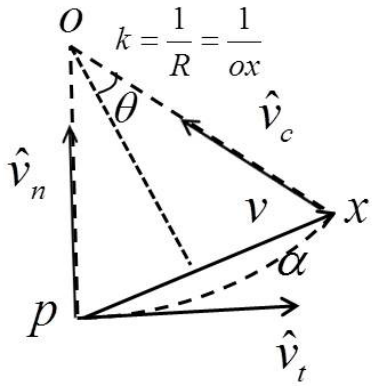

FIGURE 1. Illustration of the fundamental stick vote.

to others) on a curve. Its normal is a known unit vector $\hat{\mathbf{v}}_n$. We want to know how it influences its neighboring votee point x (a token that receives structure information from voters). Based on the previous discussion, we assume that p and x share the same structure type when we consider p is influencing x. To approximate the path of the same structure type that passes through p and x, we take the arc of the osculating circle centered at o passing through p and x as the most likely smooth path [27]. Figure 1 shows the geometric relationship between p and x. $\hat{\mathbf{v}}_t$ is the known tangent vector at p, and $\mathbf{v}$ is the vector from p to x. $\hat{\mathbf{v}}_c$ is the normal vector at x that we want to calculate. θ is the vote angle between $\mathbf{v}$ and $\hat{\mathbf{v}}_t$. We take the influence as 0 when θ is larger than $\pi/4$ for the reason that two points that have an angle larger than 90 degrees between their normals are least likely to influence each other. a is the arc length between p and x. k is the curvature of the osculating circle, which is the reciprocal of the radius $R = ox$. To calculate θ and $\hat{\mathbf{v}}_c$,

$$\theta = \arcsin \mathbf{v}^T \hat{\mathbf{v}}_n, \tag{9}$$

$$\hat{\mathbf{v}}_c = \hat{\mathbf{v}}_n \cos 2\theta - \hat{\mathbf{v}}_t \sin 2\theta. \tag{10}$$

We also add a decay profile to the tensor to model the decaying influence of the information going through the structure. Therefore, the complete unit stick vote (tensor) that encodes the normal space information received by the votee is,

$$\mathbf{V}^p = DF(a, k, \sigma)\hat{\mathbf{v}}_c\hat{\mathbf{v}}_c^T. \tag{11}$$

$DF(a, k, \sigma)$ is the decay profile that takes a, k and σ as parameters. σ is the free parameter set by the user to control the scale of voting. The decay profile can be given empirically or based on a traditional choice,

$$DF(a, k, \sigma) = e^{-(\frac{a^2+ck^2}{\sigma^2})}, \tag{12}$$

where the parameters can be derived by the geometry:

$$c = \frac{-16log(0.1)(\sigma - 1)}{\pi^2}, \tag{13}$$

$$a = \frac{\theta||\mathbf{v}||}{\sin\theta}, \tag{14}$$

$$a = \frac{2\sin\theta}{||\mathbf{v}||}. \tag{15}$$

Usually, a voter's stick vote is not a unit vector. If so, the unit tensor expressed in (11) is multiplied with the corresponding saliency of the 1-d normal space of the voter.

Likewise, when inferring structures for the 2-d normal space, we attempt to find the same normal space at the votee. In that case, 2 basis vectors are required for the 2-d normal space. The procedure is equivalent to: (1) find out the basis vectors that span voter's normal space; (2) vote or project these basis vectors, respectively, in the way the fundamental stick vote does to the votee. (3) reconstruct the complete normal space information at the votee by combining the newly generated normal space information, i.e., adding the tensor matrices created by the stick vote, respectively.

To find out the basis vectors of the voter's normal space, Mordohai proposes a method [27] that projects the voter-to-votee vector into voter's normal space and then computes the orthonormal basis vectors for the voter's normal space based on the projection and Gram-Schmidt orthogonalization procedure. This method significantly reduces the computation when calculating the basis vectors of the high dimensional normal space.

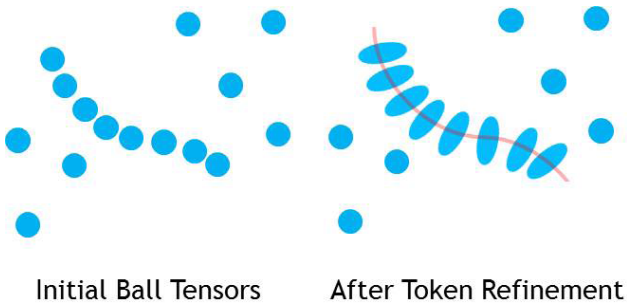

FIGURE 2. Illustration of the token refinement procedure: each point is initialized with a ball tensor; points nearby with each other form stick tensors while points far away from others remain ball tensors.

Suppose the voter's normal space is known and encoded as the tensor matrix $\mathbf{N}_d^p$, where p represents the voter point in Figure 2 and d represents the dimension of its normal space. For any fixed votee point x that receives p's vote, the voter-to-votee vector $\mathbf{v}$ is known. Then the projected vector is,

$$\mathbf{v}_n = N_d^p \mathbf{v}. \tag{16}$$

Thus, the tangent vector $\mathbf{v}_n$ is computed by

$$\mathbf{v}_t = (I - N_d^p)\mathbf{v} = \mathbf{v} - \mathbf{v}_n, \tag{17}$$

where I is the identity matrix of dimension N-by-N. Based on (17), these two vectors are then normalized as $\hat{\mathbf{v}}_n$ and $\hat{\mathbf{v}}_t$. The first constructed basis vector for the normal space is selected as $\hat{\mathbf{v}}_{n,1} = \hat{\mathbf{v}}_n$. Next, the Gram-Schmidt procedure is employed to construct the rest $d - 1$ orthonormal basis vectors $\hat{\mathbf{v}}_{n,i}$, $i = 2, 3, \ldots, d$. As a consequence, each $\hat{\mathbf{v}}_{n,i}$ is considered as the fundamental stick vote and voted to the votee as the voting procedure described above. Each stick vote results in a tensor matrix $\mathbf{V}_i^p$, $i = 1, 2, \ldots, d$ for the d-dimension normal space. Finally, these tensors are summed

up into one matrix that represents the complete information for the d-dimension normal space at votee x. The proposed method of generating the basis set facilitates the computation because for $i = 2, 3, \ldots, d$, the basis vector $\hat{\mathbf{v}}_i$ is orthogonal to $\mathbf{v}$, which means the vote angle $\theta = 0$. Hence, (10) is simplified during the computation.

C. TOKEN REFINEMENT

So far, the discussions are based on the presumption that the normal space and saliencies information are known at a given voter site. Nevertheless, in most cases, it is impossible to obtain this kind of prior knowledge. Thus, the proper initialization called token refinement, which estimates the prior information, is needed.

Figure 2 illustrates the token refinement procedure in a $2D$ space. In the token refinement procedure, each input token is initialized with a unit ball tensor indicating neither direction preference nor prior saliency information. The input tokens are then considered one by one as the voter and voted to its neighboring input tokens. In the end, all the tensors received by each input token are summed up and stored as the known normal space and saliency information. If a cluster of tokens actually belong to the same curve in the real world, and then the way they influence each other using their initial ball tensors will put major emphasis on the stick tensor, namely the 1-d normal space in a $2D$ world. As can be seen in Figure 2, the tokens along the curve influence each other during token refinement, resulting in elongating their tensors to become stick tensors in the $2D$ space, while the tokens that sparsely spread out the space receive little information from others, causing the existing ball tensors to remain.

D. TOKEN DECOMPOSITION

After token refinement, the tensor voting procedure can be completed by the method described in Section III-B. The result in the $2D$ image case is that each pixel is associated with a 2×2 matrix $\mathbf{T}$. The ultimate objective is to decide which structure type the candidate pixel should belong to. Hence, the tensor matrix $\mathbf{T}$ needs to be decomposed by (7) to extract the saliencies for the 2 structure types. In $2D$ case, it becomes:

$$\mathbf{T} = \lambda_1 \hat{\mathbf{e}}_1 \hat{\mathbf{e}}_1^T = (\lambda_1 - \lambda_2)\hat{\mathbf{e}}_1 \hat{\mathbf{e}}_1^T + \lambda_2(\hat{\mathbf{e}}_1 \hat{\mathbf{e}}_1^T + \hat{\mathbf{e}}_2 \hat{\mathbf{e}}_2^T). \quad (18)$$

If $\lambda_1 - \lambda_2 > \lambda_2 > 0$, the stick saliency is the dominant one, which indicates the certainty of one normal orientation. Therefore, the token is inferred as the part of a curve, with its estimated normal being $\hat{\mathbf{e}}_1$. If $\lambda_1 \approx \lambda_2 > 0$, the dominant component is the ball saliency, which means there is no preference of orientation. Thus, the token is estimated as the part of a region or a junction where two or more curves intersect with multiple orientations present simultaneously. Note that, if both the saliency values are very small, the candidate token is likely an outlier. This makes tensor voting capable of filtering noise.

IV. INFERENCE ALGORITHM

Based on the previous discussion, there are multiple choices with respect to how to implement the tensor voting technique. One feasible way to implement the tensor voting technique is to compute the so called voting field for each token; this method is referred to as the per-voter scheme. After token refinement procedure is complete, the per-voter algorithm examines every input token as a voter and computes the set of votes it casts to all its neighbors. That set of votes is referred as the voting field. The algorithm then integrates all the voting fields and performs tensor decomposition at each site. In [28], tensor voting is implemented based on the per-voter scheme. In addition, the algorithm is combined with the steerable filter theory [29] to rewrite the tensor voting operation as a linear combination of complex-valued convolutions, which significantly reduces the computation load.

In this paper, a straightforward implementation method of tensor voting technique referred to as the per-votee scheme is proposed to infer the human mobility trace encoded in the location data with some recordings missing. The per-voter scheme calculates one vote from point to point at one time. In order to reduce the computation, we also implement tensor voting in a sparse sense. When examining one site as a votee, we only consider the influence it received from the neighboring pixels $\{C_{ee}\}$ within the radius of approximately 3σ as reported in [10]. Furthermore, we define the sparse voting region, gather all the tensors received by each votee only in that region and decompose the result tensor matrix $\mathbf{M}_{ee}$ to determine its actual structure type. Finally, we make the voting procedure iterative so that it is able to fill the large gaps. The implementation scheme is summarized in the Algorithm 1. In the initialization stage, each voter matrix is initialized as the identity matrix $\mathbf{M}_{er}$, representing the ball tensor while each votee matrix $\mathbf{M}_{ee}$ is initialized by the zero matrix. During the voting procedure, the voter matrix $\mathbf{M}_{er}$ is firstly decomposed into two normal vectors $\{\mathbf{V}_{n1}, \mathbf{V}_{n2}\}$ which are then voted as the fundamental stick tensor to the votee. As reported in [10], each voter decomposition using (8) is computed in $O(N^3)$ time where N is the dimension of input data. And each stick vote indicated in (11) is computed in $O(N^3)$ time.

V. TRACE ANALYSIS

In this section, we carry on to analyze the trace patterns in order to discover representative behaviors or characteristics of the objects. The underlying philosophy of trace analysis is that we believe different objects' traces reflect some hidden features that can be differentiated. The objective of mining traces can vary in different contexts. For example, in our study, due to the different road layouts of distinct cities, it is highly likely that the corresponding collected human mobile traces carry different hidden features. Hence, by analyzing the properly extracted features of human mobile traces, different corresponding cities can be inferred as well as the road layout styles. Another application of trace analysis is the traffic control problem. Based on the trace analysis of each vehicle,

Algorithm 1 Tensor Voting Based on the Per-Votee Scheme

```
input 2D image with the incomplete trace points set {P_ij}, set scale of voting σ.
for r = 1, 2, ..., number of iterations do
    initialize voter matrix M_er and votee matrix M_ee.
    for i, j = 1, 2, ..., indexes of trace points do
        for each point p lies in the neighbohood of {P_ij} do
            if p ∈ {P_ij} then
                calculate DF and θ by the coordinates of the voting pair {p, P_ij};
                decompose p's M_er into 2 normal vectors {V_n1, V_n2};
                multiple {V_n1, V_n2} with DF and project them respectively to P_ij;
                collect all vectors received at P_ij and convert them into tensor.
            end
            M_ee of P_ij = sum up all tensors that P_ij has received;
        end
        update the M_er for {P_ij} by M_er = M_ee;
    end
    find out a new set of votees {C_ee} that are defined by the sparse region of {P_ij};
    set the votee matrix M_ee for {C_ee} as 2-by-2 zero matrix;
    for each point in {C_ee} do
        repeat the similar voting procedure above using the updated voter matrices.
    end
    for each pixel x of the image do
        if x ∈ {C_ee} then
            decompose the M_ee of x;
            classify the pixel x according to its saliencies;
        end
    end
    Skeletonize the updated trace points.
end
```

it is possible to identify outlier vehicles which may cause damage and loss to the traffic.

In order to perform trace analysis, proper hidden features of the complete traces have to be extracted first. Note that feature extraction depends on the specific application. Here we propose three features to be used to characterize a human mobile trace: normalized trajectory mean, Fourier descriptor and fractal dimension. These features can be categorized into two classes: spatial features (the first two) and scale-invariant feature (the fractal feature). It is worth to point out that the fractal feature provides suitable descriptions of inherent nature of the object data recorded by the GPS due to the scale-independent property brought by the fractal analysis. Hence, regardless of any resolution of the GPS measurements or any unit (meters, kilometers, etc.) that is taken to represent the trace, the fractal feature stays consistent.

To ease the illustration of feature extraction procedure, we assume the trace data are in 2*D*. Let the set of inferred trace points be $\{T\}$, and we assume there are n points that continuously form the inferred trace, each with two coordinates in the 2*D* space: $(T_x^i, T_y^i), i = 1, 2, \ldots, n$. The normalized trajectory mean is defined as:

$$\mathbf{m} = \frac{1}{n}\sum_{i=1}^{n}(T_x^i, T_y^i) - (min(T_x^i), min(T_y^i)). \tag{19}$$

The normalized trajectory mean is to roughly indicate the cover range or off-set range of the trace. Sometimes the trace might form a loop and therefore, the Fourier descriptor is needed to characterize the trace in this case. For the coordinates of each trace point, there is a corresponding complex value formed as:

$$z^{i-1} = T_x^i + j \cdot T_y^i, \tag{20}$$

where j is the imaginary unit. Thus, we can obtain a sequence of $\{z^i\}$ and the discrete Fourier transform of $\{z^i\}$ is:

$$a(u) = \frac{1}{n}\sum_{i=0}^{n-1} z^i e^{-j2\pi ui/n}, u = 0, 1, \ldots, n-1. \tag{21}$$

The complex coefficient $a(u)$ is referred to as the Fourier descriptor and its absolute value indicates the magnitude of corresponding frequency component. Note that the Fourier descriptor is independent of how the first trace point is chosen. High frequency components present rapidly varying details in the trace while low frequency components determine the shape of trace at the coarse level. Denote the absolute value of low frequency components and high frequency components as a_L and a_H, respectively. We then define R_f to be the ratio of low vs high frequency components:

$$R_f = a_L/a_H, \tag{22}$$

which serves as the Fourier descriptor feature for the further trace analysis.

In order to include scale-independent feature, we briefly introduce the fractal dimension to the feature extraction procedure. The underlying philosophy of fractal dimension is that self-similarity and repeating patterns exist in nature objects. A fractal is a shape made of parts similar to the whole in some way. Usually, a fractal is a set of objects that has a fractal dimension that exceeds its topological dimension. The fractal dimension indicates the effect of space occupation by the complex shape. There are many ways to formulate fractal dimension, such as the Hausdorff dimension, the Richardson Law [22] and so forth. Here we adopt the Richardson Law to define fractal dimension as:

$$L(\epsilon) \propto \epsilon^{D-D_f}, \tag{23}$$

where D is the topological dimension of the trace and $D = 1$. D_f is the fractal dimension. ϵ is the imaginary unit "rule"

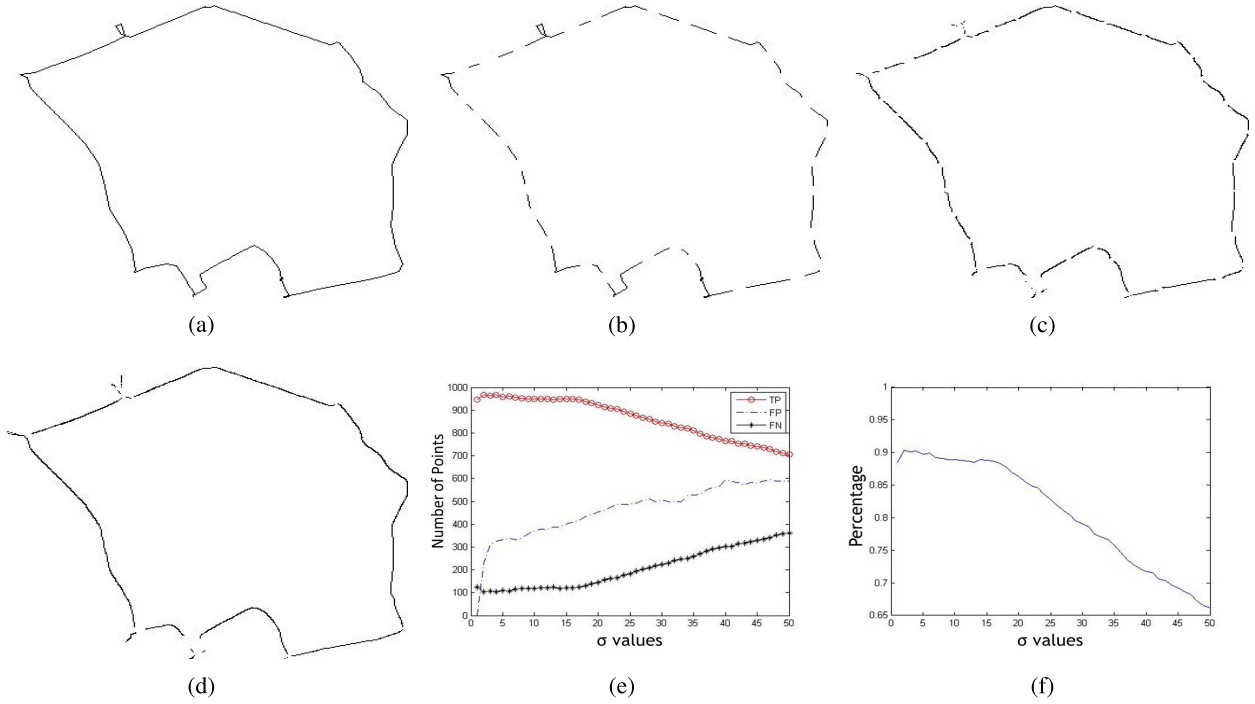

FIGURE 3. Simulation results. (a) One instance of complete human mobility trace converted from the GPS data. (b) Corresponding sampled human mobility trace with missing segments. (c) Inferred result employing tensor voting with voting scale $\sigma = 1$. (d) Inferred result employing tensor voting with voting scale $\sigma = 2$. (e) Corresponding true positive, false positive and false negative curves. (f) Corresponding true positive ratio curve.

used to measure the trace. It can be a line segment for measuring curves, or a square/circle for measuring planar objects, or a cube/ball for measuring solid objects. $L(\epsilon)$ is the number of "rulers" used to continuously cover the entire trace. By taking the log of (23), we have

$$\ln(L(\epsilon)) = (1 - D_f) \ln \epsilon. \tag{24}$$

Therefore, we can vary the unit "ruler" ϵ and compute the corresponding counts $(L(\epsilon))$ of the trace. The fractal dimension D_f is then calculated by linear regression via (24).

So far we have obtained all the features to characterize a trace: the normalized trajectory mean $\mathbf{m}$, the Fourier descriptor feature R_f, and the fractal dimension D_f. These extracted features of traces are then input into our classification model to infer the different corresponding cities. We choose the logistic regression model for our trace classification/city inference task. Since the classification model is not the key point in our paper, we refer readers to the work in [30] for further information regarding the classification model used here.

VI. EXAMPLE AND ANALYSIS

The performance of the proposed tensor voting algorithm is verified through extensive experiments on the human mobility data collected in New York city by the GPS [31]. Each of the 39 collected traces is firstly converted into $2D$ binary images of the dimension 314×351. The $2D$ images are then randomly sampled to have missing or broken segments of approximately 7-pixel length on average. Finally, the proposed tensor voting algorithm is applied to the images with missing parts and outputs the inferred complete trace of the human object. Due to the small size of missing gaps, the iteration number is set fixed to 1 in this paper for simplicity. For the cases where there are large missing gaps, the iteration number should be set larger when the adjacent iterations return sufficiently similar results. The experiments are conducted extensively to each image with various values of σ, which is the only free parameter that controls the scale of voting, ranging from 1 to 50. One instance of the complete traces and corresponding sampled trace with missing parts are shown in Fig. 3 (a) and (b), respectively. Fig. 3 (c) and (d) are the inferred traces by setting the scale of voting parameter $\sigma = 1$ and $\sigma = 2$, respectively. As can be seen in these two figures, the gaps are not fully connected using too small voting scale because the structure elements are unable to influent further points. While in the situation that σ is too large, the gaps can be connected however there are many over-detected segments of the traces, which make the performance inefficient. One particularly interesting point is that the algorithm even recovers the missing arcs appropriately. During the whole procedure of tensor voting algorithm, there is no user effort required to be input to indicate which part of the trace is missing and should be inferred. The inference is completed fully automatically once the only free parameter σ is fixed. This is one aspect of the strength of tensor

voting method. The key of implementing tensor voting algorithm is the determination of data-driven parameters: the number of iterations and the scale of voting σ. Empirically, larger values of iterations and σ will make the algorithm capable of filling larger missing gaps, while the smaller values of them are more suitable for the traces of plenty of narrow missing gaps.

There are massive efforts contributed to the performance measure for the tracking problems [32]–[34]. To validate the inferred traces and quantify the performance of the proposed algorithm, the similarity metrics proposed in [35] is adopted due to its practical merits. Denote the complete trace points of the ground truth as the set $\{S\}$ while the set of inferred trace points being regarded as set $\{T\}$ as defined in previous sections. The true positive (TP) of the inference algorithm is defined as the cardinality of the set $\{p\}_i$ satisfying that:

$$\{p\}_i = \{p | p \in \{T\}, p \in \{S\}\}. \tag{25}$$

The false positive (FP) and false negative (FN) of the inference algorithm are defined in the similar fashion:

$$FP = Card(\{p\}_j); \{p\}_j = \{p | p \notin \{T\}, p \in \{S\}\}, \tag{26}$$

$$FN = Card(\{p\}_k); \{p\}_k = \{p | p \in \{T\}, p \notin \{S\}\}. \tag{27}$$

Since the trace points only occupy a small portion of the entire image, it would be irrational to take the true negative (TN) into account when evaluating the accuracy of the algorithm. In addition, due to the specific setting of the tracking problem, most interests have been focused on the value of TP. Furthermore, in order to compare the performance on various traces, we compute the TP ratio by dividing the TP values by the cardinality of the set $\{S\}$. The quantified performance of the tensor voting algorithm with various σ values is shown in Fig. 3(e)-(f). The importance is placed on the TP curve due to the specific settings of the tracking problem. And the optimal TP ratio achieves 90.36% at $\sigma = 2$ while the numbers of FP and FN are relatively small. Considering the average lenght of the missing segments in our experiments is 7-pixel, this result is consistent with the theory represent in [10], which states that the points within the circle centered at a voter of a radius of approximate 3σ would effectively receive the votes. When the σ is chosen too small, the points are under little influence from others, leading to the failure of inferring large gaps. On the other hand, as the σ grows larger, the points far away with each other interfere so much that the reconstructed results are full of noise. The small fluctuations present locally in the four curves are consequence of different lengths of missing segments that appear in the input images.

To show the advantage of proposed sparse tensor voting algorithm, we also construct the control group or so called victim image using other approach to accomplish the inference task. In this study, we choose to directly connect each pair of the end points of the missing segments, i.e. the control approach assumes the missing parts are all linear segments, which will fail where the true trace exhibits curves. The performance comparison between the proposed tensor voting (TV) algorithm and control method is shown in the Table 1, where the experiments are both conducted on the same trace. It can be seen from the Table 1 that the proposed algorithm outperforms the control one since the missing segments are not all linear segments. In the situation where most miss segments are curves, the performance difference between the two methods will be more significant.

TABLE 1. Performance comparison between the proposed tensor voting algorithm and victim method.

Method	TP	TP ratio	FP	FN
TV	966	90.36%	230	103
Victim	887	82.97%	293	182

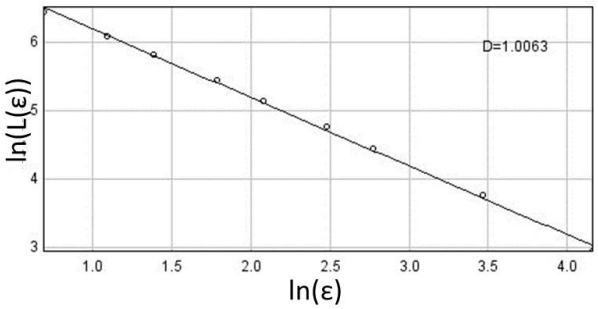

FIGURE 4. Fractal dimension computation via $\ln(L(\epsilon))$ vs $\ln \epsilon$ plot, corresponding to the same trace used in Fig. 3.

To evaluate the proposed methods for trace analysis, we first present the illustrations of extracting fractal dimensions for each trace. The traces are converted into binary images which are then computed using (24) with the varying $\epsilon = 2, 3, 4, 6, 8, 12, 16, 32, 64$ in the unit of pixels. In Fig. 4, we illustrate the fractal dimension calculated by linear regression for the same trace used in Fig. 3. As can be seen in Fig. 4, the fractal dimension of the investigating trace is larger than 1, indicating that the shape of the trace is "more complex" than a straight line in 2D. For the Fourier descriptor feature R_f, we take the average of first 30% portion of the frequency components in (21) as the low frequency components a_L and the average of last 30% portion of the frequency components as the high frequency components a_H. The feature R_f is then computed according to (22). As indicated in Section V, we feed the logistic regression model with the extracted features: the normalized trajectory mean $\mathbf{m}$, the Fourier descriptor feature R_f, and the fractal dimension D_f to classify the traces into different cities. We perform the trace analysis on two data sets with 39 traces from New York (label 0) and 41 traces from Orlando (label 1), respectively. The training and testing data sets are formed by 4-fold cross validation, each time with 60 traces in the training set and 20 traces in the testing set where the traces are divided randomly. The precision of regression results are obtained by averaging each cross

TABLE 2. Trace analysis via logistic regression.

Training set	Estimated label 0	Estimated label 1
True label 0	26	4
True label 1	8	22
Testing set	**Estimated label 0**	**Estimated label 1**
True label 0	7	2
True label 1	1	10

validation result, and are shown as the confusion matrices in Table 2 where the numerical values refer to the counts of correcly/incorrectly classified traces. As can be seen in Table 2, the precision is 80% in the training set and 85% in the testing set, which demonstrates the effectiveness of our trace analysis methods.

VII. CONCLUSIONS AND FUTURE WORK

This work provides an effective approach to infer the human mobility trace as the key object frequently utilized in networking, given that the observed location data exist missing parts. Traditional tracking techniques seldom deal with the problem of missing data setting. By employing the tensor voting technique as one of the artificial intelligence methods, the trace inference is done in an automatic fashion. The only free parameter that requires the user input is the voting scale. One advantage of the proposed algorithm is that there is no requirement of user instructions for identifying which part to inferred. The algorithm discovers the missing positions and accomplishes inference. The sparse per-votee implementation scheme significantly reduce the computation, making the algorithm suitable for potentially large-scale data set and online analysis. By employing the similarity metrics between the inferred trace and the ground truth, our algorithm shows its power in recover the human mobility trace accurately. The tensor grouping method can be applied to estimate local dimension in manifold learning or function approximation tasks. The fractal analysis can be designed for image compression, denoising and channel estimation. Future work may also involve with: (1) modifying decay profile to better fit specific problems; (2) combining tensor representations with other information, for instance, combining the second order tensor voting with the first order information as the polarity coefficient to detect end point, which prevents over-detecting points.

REFERENCES

[1] R. Sanchez Grandia, V. Aucejo Galindo, A. Usieto Galve, and R. Vives Fos, "General formulation for magnetic forces in linear materials and permanent magnets," *IEEE Trans. Magn.*, vol. 44, no. 9, pp. 2134–2140, Sep. 2008.

[2] Q. Li and D. Schonfeld, "Multilinear discriminant analysis for higher-order tensor data classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 12, pp. 2524–2537, Dec. 2014.

[3] F. T. Arslan and A. M. Grigoryan, "Fast splitting α-rooting method of image enhancement: Tensor representation," *IEEE Trans. Image Process.*, vol. 15, no. 11, pp. 3375–3384, Nov. 2006.

[4] R. Moreno, M. A. Garcia, D. Puig, L. Pizarro, B. Burgeth, and J. Weickert, "On improving the efficiency of tensor voting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 11, pp. 2215–2228, Nov. 2011.

[5] M.-H. Zayani, V. Gauthier, I. Slama, and D. Zeghlache, "Tensor-based link prediction in intermittently connected wireless networks," *Comput. Res. Repository CoRR*, vol. abs/1108.2606, 2011.

[6] L. Liu, Z. Han, Z. Wu, and L. Qian, "Collaborative compressive sensing based dynamic spectrum sensing and mobile primary user localization in cognitive radio networks," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Houston, TX, USA, Dec. 2011, pp. 1–5.

[7] K. Lee, S. Hong, S. J. Kim, I. Rhee, and S. Chong, "SLAW: A new mobility model for human walks," in *Proc. INFOCOM*, Rio de Janeiro, Brazil, Apr. 2009, pp. 855–863.

[8] I. Rhee, M. Shin, S. Hong, K. Lee, S. J. Kim, and S. Chong, "On the Levy-walk nature of human mobility," *IEEE/ACM Trans. Netw.*, vol. 19, no. 3, pp. 630–643, Jun. 2011.

[9] E. Pan, M. Pan, Z. Han, and V. Wright, "Mobile trace inference based on tensor voting," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Austin, TX, USA, Dec. 2014, pp. 4891–4897.

[10] B. J. King, "Range data analysis by free-space modeling and tensor voting," Ph.D. dissertation, Rensselaer Polytech. Inst., Troy, NY, USA, 2008.

[11] M. Reisert and H. Burkhardt, "Efficient tensor voting with 3D tensorial harmonics," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Anchorage, AK, USA, Jun. 2008, pp. 1–7.

[12] G. Guy and G. Medioni, "Inferring global perceptual contours from local features," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, New York, NY, USA, Jun. 1993, pp. 786–787.

[13] J. Kang, I. Cohen, and G. Medioni, "Continuous multi-views tracking using tensor voting," in *Proc. Workshop Motion Video Comput.*, Orlando, FL, USA, Dec. 2002, pp. 181–186.

[14] P. Kornprobst and G. Medioni, "Tracking segmented objects using tensor voting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 2. Hilton Head Island, SC, USA, Jun. 2000, pp. 118–125.

[15] A. Narayanaswamy, Y. Wang, and B. Roysam, "3-D image preprocessing algorithms for improved automated tracing of neuronal arbors," *Neuroinformatics*, vol. 9, nos. 2–3, pp. 219–231, Sep. 2011.

[16] L. Li and D. Boulware, "High-order tensor decomposition for large-scale data analysis," in *Proc. IEEE Int. Congr. Big Data (BigData Congress)*, New York, NY, USA, Jun. 2015, pp. 665–668.

[17] A. P. Liavas and N. D. Sidiropoulos, "Parallel algorithms for large scale constrained tensor decomposition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, South Brisbane, QLD, Australia, Apr. 2015, pp. 2459–2463.

[18] M. Mardani, G. Mateos, and G. B. Giannakis, "Subspace learning and imputation for streaming big data matrices and tensors," *IEEE Trans. Signal Process.*, vol. 63, no. 10, pp. 2663–2677, May 2015.

[19] N. Vervliet, O. Debals, L. Sorber, and L. De Lathauwer, "Breaking the curse of dimensionality using decompositions of incomplete tensors: Tensor-based scientific computing in big data analysis," *IEEE Signal Process. Mag.*, vol. 31, no. 5, pp. 71–79, Sep. 2014.

[20] N. Anjum and A. Cavallaro, "Multifeature object trajectory clustering for video analysis," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 11, pp. 1555–1564, Nov. 2008.

[21] J.-G. Lee, J. Han, and X. Li, "A unifying framework of mining trajectory patterns of various temporal tightness," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 6, pp. 1478–1490, Jun. 2015.

[22] D. Mouillot and D. Viale, "Satellite tracking of a fin whale (balaenoptera physalus) in the north-western Mediterranean Sea and fractal analysis of its trajectory," *Hydrobiologia*, vol. 452, nos. 1–3, pp. 163–171, Jun. 2001.

[23] D. Zhang and J. P. G. Sterbenz, "Robustness analysis of mobile ad hoc networks using human mobility traces," in *Proc. 11th Int. Conf. Design Rel. Commun. Netw. (DRCN)*, Kansas City, MO, USA, Mar. 2015, pp. 125–132.

[24] H. Liu and J. Li, "Unsupervised multi-target trajectory detection, learning and analysis in complicated environments," in *Proc. 21st Int. Conf. Pattern Recognit. (ICPR)*, Tsukuba, Japan, Nov. 2012, pp. 3716–3720.

[25] J.-G. Ko and J.-H. Yoo, "Rectified trajectory analysis based abnormal loitering detection for video surveillance," in *Proc. 1st Int. Conf. Artif. Intell., Modelling Simulation (AIMS)*, Kota Kinabalu, Malaysia, Dec. 2013, pp. 289–293.

[26] C. T. Zahn, "Graph-theoretical methods for detecting and describing gestalt clusters," *IEEE Trans. Comput.*, vol. C-20, no. 1, pp. 68–86, Jan. 1971.

[27] P. Mordohai and G. Medioni, *Tensor Voting: A Perceptual Organization Approach to Computer Vision and Machine Learning* (Synthesis Lectures on Image, Video, and Multimedia Processing). San Rafael, CA, USA: Morgan & Claypool Publishers, Nov. 2006.
[28] E. Franken, M. van Almsick, P. Rongen, L. Florack, and B. ter Haar Romeny, "An efficient method for tensor voting using steerable filters," in *Proc. 9th Eur. Conf. Comput. Vis. (ECCV)*, 2006, pp. 228–240.
[29] W. T. Freeman and E. H. Adelson, "The design and use of steerable filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 9, pp. 891–906, Sep. 1991.
[30] B. Krishnapuram, L. Carin, M. A. T. Figueiredo, and A. J. Hartemink, "Sparse multinomial logistic regression: Fast algorithms and generalization bounds," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 6, pp. 957–968, Jun. 2005.
[31] I. Rhee, M. Shin, S. Hong, K. Lee, S. Kim, and S. Chong. (Jul. 2009). *CRAWDAD Data Set NCSU/Mobilitymodels (v. 2009-07-23)*. [Online]. Available: http://crawdad.org/ncsu/mobilitymodels/
[32] A. A. Gorji, R. Tharmarasa, and T. Kirubarajan, "Performance measures for multiple target tracking problems," in *Proc. 14th Int. Conf. Inf. Fusion (FUSION)*, Chicago, IL, USA, Jul. 2011, pp. 1–8.
[33] C. J. Needham and R. D. Boyle, "Performance evaluation metrics and statistics for positional tracker evaluation," in *Proc. 3rd Int. Conf. Comput. Vis. Syst. (ICVS)*, Berlin, Germany, 2003, pp. 278–289.
[34] F. Yin, D. Makris, and S. A. Velastin, "Performance evaluation of object tracking algorithms," in *Proc. 10th IEEE Int. Workshop Perform. Eval. Tracking Surveill. (PETS)*, Rio de Janeiro, Brazil, Oct. 2007, pp. 1–8.
[35] L. M. Brown *et al.*, "Performance evaluation of surveillance systems under varying conditions," in *Proc. IEEE PETS Workshop*, Jan. 2005, pp. 1–8.

ERTE PAN (S'14) received the B.E. degree in electrical and computer engineering from Wuhan University, China, in 2010. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Houston. He has been a Research Assistant with the Wireless Networking, Signal Processing and Security Laboratory since 2013. His research interests include non-parametric inference, deep learning networks, big data analysis, and sublinear methods on smart grid networks. He served as a Peer Reviewer of the IEEE TRANSACTIONS ON MEDICAL IMAGING, the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, and the IEEE TRANSACTIONS ON SMART GRID.

MIAO PAN (S'07–M'12) received the B.Sc. degree in electrical engineering from the Dalian University of Technology, China, in 2004, the M.A.Sc. degree in electrical and computer engineering from the Beijing University of Posts and Telecommunications, China, in 2007, and the Ph.D. degree in electrical and computer engineering from the University of Florida, in 2012. He is currently an Assistant Professor with the Electrical and Computer Engineering Department, University of Houston, TX. His research interests include cognitive radio networks, cyber-physical systems, and cybersecurity.

ZHU HAN (S'01–M'04–SM'09–F'14) received the B.S. degree in electronics engineering from Tsinghua University, in 1997, and the M.S. and Ph.D. degrees in electrical engineering from the University of Maryland, College Park, in 1999 and 2003, respectively. From 2000 to 2002, he was an R&D Engineer with JDS Uniphase Corporation, Germantown, MD. From 2003 to 2006, he was a Research Associate with the University of Maryland. From 2006 to 2008, he was an Assistant Professor with Boise State University, ID. He is currently a Professor with the Electrical and Computer Engineering Department and the Computer Science Department, University of Houston, TX. His research interests include wireless resource allocation and management, wireless communications and networking, game theory, wireless multimedia, security, and smart grid communication. He received an NSF Career Award in 2010, the Fred W. Ellersick Prize of the IEEE Communication Society in 2011, the EURASIP Best Paper Award for the *Journal on Advances in Signal Processing* in 2015, and several best paper awards in the IEEE conferences, and is an IEEE Communications Society Distinguished Lecturer