# Distributed Sensor Perception via Sparse Representation

Allen Y. Yang, *Member, IEEE,* Michael Gastpar, *Member, IEEE,* Ruzena Bajcsy, *Fellow, IEEE* and S. Shankar Sastry, *Fellow, IEEE*

***Abstract*—Sensor network scenarios are considered where the underlying signals of interest exhibit a degree of sparsity, which means that in an appropriate basis, they can be expressed in terms of a small number of nonzero coefficients. Following the emerging theory of compressive sensing, an overall architecture is considered where the sensors acquire potentially noisy projections of the data, and the underlying sparsity is exploited to recover useful information about the signals of interest, which will be referred to as distributed sensor perception. First, we discuss the question of which projections of the data should be acquired, and how many of them. Then, we discuss how to take advantage of possible joint sparsity of the signals acquired by multiple sensors, and show how this can further improve the inference of the events from the sensor network. Two practical sensor applications are demonstrated, namely, distributed wearable action recognition using low-power motion sensors and distributed object recognition using high-power camera sensors. Experimental data support the utility of the compressive sensing framework in distributed sensor perception.**

## I. Introduction

In the last decade, the information technology industry continues to advance on multiple scientific fronts, including integrated circuit design, wireless communication, and heterogeneous sensor technologies. Recent progress in more powerful mobile processors and wireless devices has empowered new applications in *wireless sensor networks* (WSNs) that differentiate themselves from traditional low-power sensor applications in the past, such as simple detection and registration of temperature, precipitation, and sound. For instance, today many mobile phones possess considerable computation and communication capabilities. Often, these devices also retain rich sensing components to interact with the environment and human users, including cameras, microphones, positioning sensors, and motion sensors. In industrial surveillance, multiple wireless devices with heterogeneous sensing capabilities can be configured in a network to monitor the environmental information in factories. In intelligent transportation, stationary and mobile sensor networks have been used to support real-time traffic surveillance and autonomous driving.

A WSN usually consists of a set of sensor nodes and one or more base-station computers. A wireless sensor node is often called a mote, which is an integrated device consisting of sensing, data processing, and communication components. Stationary motes can be deployed both indoors and outdoors. Mobile motes can be instrumented on humans or air/ground vehicles. As shown in Figure 1, these motes can communicate among each other via wireless channels, and also communicate with base stations as gateways and output the sensor data for processing in higher-level applications. The reader is referred to [1]–[3] for more detailed surveys about the literature of WSNs.

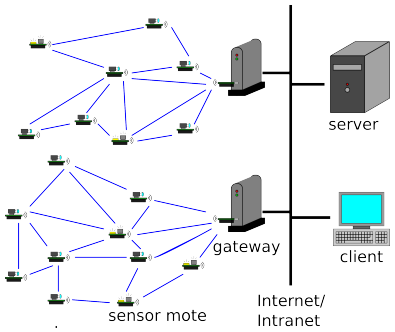


Fig. 1. A typical architecture of WSNs.

The infrastructure of WSNs can provide great benefits to their applications. Possibly the most important benefit is that, with mobile processors and memory directly integrated with sensors, certain computation can be "pushed" to the edge of the networks for faster decision making for time-critical applications. In addition, wireless communication between sensors and to the base station enables miniature sensors to be rapidly deployed in complex indoor or outdoor terrain. Furthermore, sensor networks can fuse measurements from a wide spectrum of sensing modalities.

However, these advantages cannot come without sacrifices on the resources allocated for WSNs. The fundamental constraint for a wireless sensor is its limited power supply, typically from portable batteries integrated as part of the sensor node. Assuming a WSN is intended to function over a prolonged period of time, it dictates that the hardware implementation of the sensor node can only provide *limited computational power* and *limited communication bandwidth*.

Among many important problems associated with analyzing sensor networks (such as hardware design, communication channels, and security, etc.), in this paper, we are interested in estimation and recognition of certain physical events that are observed within the setting of a WSN, which is referred to as *distributed sensor perception*. Applications in distributed sensor perception must answer a quintessential question: *How*

A. Yang, M. Gastpar, R. Bajcsy, and S. Sastry are with the Department of EECS, University of California, Berkeley, CA, 94720 USA e-mail: {yang,gastpar,bajcsy,sastry}@eecs.berkeley.edu.

This work is supported in part by ARO MURI W91INF-06-1-0076, ARL MAST-CTA W91INF-08-2-0004, and the NSF TRUST Center.

*to design a sensor network system such that its performance in sensing and perception surpasses simply the sum of its individual parts?* More specifically, distributed sensor perception concerns the following fundamental problems:

1) How does an algorithm effectively harness the distributed nature of sensor networks to detect and recognize events of interest?
2) How does an algorithm address the robustness issue in the presence of moderate data noise and outliers?
3) How does an algorithm adapt to on-the-fly changes in the network configuration?

The focus of the paper, under the overarching theme of this special issue, is to investigate the rich phenomena of *sparsity* that are often exhibited in distributed sensor signals, and to showcase how one can take advantage of the emerging theory of *compressive sensing* (CS) in searching for elegant solutions to the above questions.

The paper intends to present a hands-on survey about the state-of-the-art research results broadly related to WSNs and CS. Although the concept of sparse representation in sensor networks is still quite abstract at this point, an investigator who would like to design a sensor system to solve a practical problem at hand must have a clear understanding about at least the following two components: First, what sampling functions the system should employ to measure the physical events on the sensor side; Second, what inferencing functions the system should design on the base-station side to accurately reconstruct and represent the events of interest. The paper will guide the reader step-by-step in seeking these answers.

## II. Background

We first start with a brief overview of the basic CS theory. The reader can find more thorough treatment of the theory in [4]–[6]. In general, a signal $\tilde{\boldsymbol{x}} \in \mathbb{R}^n$ is considered sparse if most of its coefficients $\boldsymbol{x}_0$ under an appropriate basis $\Psi$ are zero:

$$\tilde{\boldsymbol{x}} = \Psi \boldsymbol{x}_0, \tag{1}$$

where $k = \|\boldsymbol{x}_0\|_0$ is called the sparsity of $\boldsymbol{x}_0$. Sparsity and many of its applications have been extensively studied in the past. Arguably, one of the most popular applications of sparse representation is in image compression, where a 2-D image with dense (nonzero) pixel values can be encoded and compressed using a small fraction of the coefficients after a linear transformation. In this example, the transformation $\Psi$ may represent a discrete cosine transform (DCT) basis or a wavelet basis.

Compressive sensing (CS) has been motivated by a striking observation: If the source signal $\boldsymbol{x}_0 \in \mathbb{R}^n$ is sufficiently sparse, with high probability, $\boldsymbol{x}_0$ can be recovered from a smaller set of observations $\boldsymbol{y} \in \mathbb{R}^d$ under a linear projection on $\tilde{\boldsymbol{x}}$:

$$\boldsymbol{y} = A\tilde{\boldsymbol{x}} = A\Psi\boldsymbol{x}_0, \tag{2}$$

where the sensing matrix $A \in \mathbb{R}^{d \times n}$ is typically full-rank with $d < n$.

In (2), the columns of the sensing matrix $A$ constitute an overcomplete dictionary, and $\boldsymbol{y}$ lies in a lower-dimensional space than $\boldsymbol{x}_0$. Therefore, there exist infinitely many solutions of $\boldsymbol{x}$ that give rise to $\boldsymbol{y}$. The theory of CS states that, for most full-rank matrices $A$ that are *incoherent* to $\Psi$, if $\boldsymbol{x}_0$ is sparse with respect to its dimension $n$, it is the unique solution of a regularized $\ell_0$-minimization ($\ell_0$-min) program [7]:

$$\min \|\boldsymbol{x}\|_0 \text{ subject to } \boldsymbol{y} = A\Psi\boldsymbol{x}. \tag{3}$$

Unfortunately, $\ell_0$-min is an NP-hard problem, and solving for the optimal solution basically requires an expensive combinatorial search over all possible combinations of nonzero coefficients. Hence, the bulk of study in CS involves determining a nontrivial equivalence relationship that provides a theoretical guarantee: If the true solution $\boldsymbol{x}_0$ is *sufficiently* sparse, $\boldsymbol{x}_0$ can be efficiently recovered by a more tractable $\ell_1$-minimization ($\ell_1$-min):

$$\min \|\boldsymbol{x}\|_1 \text{ subject to } \boldsymbol{y} = A\Psi\boldsymbol{x}. \tag{4}$$

This relationship is conveniently called $\ell_0/\ell_1$ equivalence [5], [8]. The literature of convex optimization has provided a long list of solvers for this task, such as *orthogonal matching pursuit* (OMP) [9], *basis pursuit* (BP) [10], *least angle regression* (LARS) [11], and the *LASSO* [12].

The phenomena of sparsity are abundant in sensor networks. For camera sensors, a poster example is the so-called *single-pixel camera* [13]. Traditional imaging mechanisms require expensive sensing arrays and memory to store 2-D image pixels in full resolution, only to be reduced to a small portion of (nonzero) coefficients later in the compression stage. In contrast, a single-pixel camera sequentially samples one pixel at a time, each of which is a random linear projection of the original image pixels. In the compressive sensing formulation (2), each sequential observation becomes a scalar coefficient in $\boldsymbol{y}$, and a random linear projection is represented by a row vector in the sensing matrix $A$. To recover the image pixels $\tilde{\boldsymbol{x}}$ from $\boldsymbol{y}$, the decoder should choose a proper sparsity basis $\Psi$ (e.g., the Fourier basis or the wavelet basis), and call upon the $\ell_1$-min algorithm (4) to recover the sparse coefficients $\boldsymbol{x}_0$.

The idea of single-pixel camera captures a unique benefit of CS in sensor networks: In resource-constraint systems, if high-dimensional observations exhibit certain sparsity in either the spatial or frequency domain, CS provides a means to simultaneously *sense* and *compress* the data using just matrix-vector multiplication at the edge of the network. Subsequently, the dominant complexity in computation to decode the original data is transferred to the decoder on the base station that often has much higher computational power.

Applying the principles of CS in a distributed sensor network naturally raises two questions: First, on each sensor node, how should one properly choose a good sensing matrix $A$ based on the characteristics of the sensor measurements, and what is a good projection dimension $d$ to guarantee a $\ell_1$-min algorithm can later recover the high-dimensional sparse signals? Second, on the base-station side, if a physical event is observed in multiple instances by sensors at different locations or the same sensor over time, how can one take advantage of the possible joint sparsity among multiple sensor observations and improve the accuracy in inferencing the event from the network? These are the questions we intend to answer.

The rest of the paper is organized as the following. Section III discusses sensing matrices for distributed sensors and their individual performance bounds; Section IV formulates the concept of joint sparsity and discusses strategies to implement global inferencing algorithms over the network.

## III. Random Projections: A Universal Dimensionality-Reduction Scheme

An unconventional result in CS is that, in high dimensional spaces, random projections can be a universal sampling operation to encode sparse signals in an appropriate basis. We mentioned earlier that an important property for a good sensing matrix $A$ in (4) is that $A$ must be sufficiently *incoherent* to the basis $\Psi$ under which the signal is sparse [6], [8], [14].

To define random projections, a standard approach considers a matrix $A$ whose entries $a_{ij}$ are drawn from an *independent and identically distributed* (i.i.d.), zero-mean Gaussian distribution. In practice, the random coefficients $a_{ij}$ are generated by a pseudo-random number generator. Furthermore, due to a practical concern that most current low-power mobile processors only support fixed-point instructions, another projection matrix is often used called the Rademacher random matrix, whose entries are assigned to be only $\pm 1$ with equal probability. After the projection, each scalar sample $y_i = [a_{i1}, \cdots, a_{in}] \cdot \boldsymbol{x}_0$ is a random combination of the sensor measurements $\boldsymbol{x}_0$.

Depending on the nature of applications, in fact, many other sensing matrices have been studied aside from random projections. For instance, in image compression, several papers in the past have studied star-shape Fourier sampling [15], random partial Fourier matrices [16], and scrambled block Hadamard ensembles [17]. These sensing matrices are all designed to cater to a particular set of sparse signals, and hence, they generally would perform better in recovering sparsity in CS than random projections [18].

On the other hand, random projections as a universal encoding strategy [8] do not depend on specific knowledge about the source signals. This is particularly relevant to applications in sensor networks, where a wireless network may support both high-power imaging sensors and other low-power sensors, and a wide range of inference functions may not be identified at the time of deployment. More specifically, random projections hold the following advantages:

1) *Universal incoherence*. Random matrices $A$ can be coupled with most conventional sparsity bases $\Psi$ such that, with high probability, sparse signals can be recovered by efficient solvers, such as $\ell_1$-min on the projected measurements $\boldsymbol{y}$.
2) *Data independence*. The construction of a random matrix does not depend on any prior data from the application. In fact, given an explicit pseudo-random number generator, the sensors and the base station only need to agree on a single random seed to generate the same random matrices of any dimension.
3) *Robustness*. Transmission of randomly projected coefficients is robust to packet loss in the network. Even if part of the coefficients in $\boldsymbol{y}$ is lost, the receiver can still reconstruct a partial random matrix $A$ and recover the sparse signal at the expense of less accuracy. Another strategy to improve robustness is to progressively sample the source signal using random projections until the accuracy of the reconstruction exceeds a certain threshold.

In this section, we will mainly focus on using random projections as sensing matrices. One question we will discuss in depth is: How many random projections $d$ have to be acquired in order to attain good performance? This question is particularly interesting when the acquired random projections are subject to additional noise, for example due to non-idealities in the observation process or due to subsequent compression. In the sequel, we provide a brief overview of the state of the art regarding the necessary number of samples. For clarity, in this paper, we often assume in our formulation (4) that $\Psi$ is an identity matrix $I$ without loss of generality.

### A. Exact Recovery

Let us first consider the requirement to exactly recover the original sparse signal $\boldsymbol{x}_0$. From elementary linear algebra it is clear that at least $d \geq k+1$ samples must be acquired; otherwise, some of the $k$-dimensional subspaces spanned by $k$ columns of $A$ must coincide, and hence, exact recovery is not feasible. For nonzero sparsity rate $\rho = \lim_{n\to\infty} k/n$, it is also instructive to write this in terms of a *sampling rate* $\delta = \lim_{n\to\infty} d/n$, meaning that the necessary condition becomes

$$\delta \quad \geq \quad \rho. \tag{5}$$

This necessary condition still allows the subspaces corresponding to different $k$-dimensional subsets of the columns of A to coincide in $k-1$ or fewer dimensions. When the sparsity coefficients are drawn randomly from a continuous distribution, this is not an issue since the probability that the samples come to lie in this intersection is zero. However, if one wants to require all subspaces to be distinct (and intersect only at the origin), then a necessary condition is [7]

$$\delta \quad \geq \quad 2\rho. \tag{6}$$

In order to attain these lower bounds, no efficient algorithms are known and it appears that one has to resort to exhaustive search over all possible $\binom{n}{k}$ sparse supports. However, a key result of CS is that if further samples are acquired, then polynomial-complexity algorithms exist (e.g., the aforementioned $\ell_0/\ell_1$ equivalence). A sufficient condition for this is to acquire[1]

$$\delta \quad \geq \quad O\left(k/n \log(n/k)\right) \tag{7}$$

random projections. For the special case where $A$ is a Gaussian random matrix, the precise scaling constants have been found [19]. However, the same constants are not currently known in other cases. It is interesting to observe that this still corresponds to a *finite* sampling rate, albeit potentially considerably larger than the fundamental lower bound.

[1] "$f = O(g)$" means function $f$ is bounded from above by $g$ asymptotically. "$f = \Theta(g)$" means $f$ is bounded from both above and below by $g$ asymptotically.

### B. Recovery with small $\ell_2$ distortion

When noise is added to the samples, generally it will not be possible to exactly recover the original signal $\boldsymbol{x}_0$. The noisy random projections are given by

$$\boldsymbol{y} = A\boldsymbol{x}_0 + \boldsymbol{e}, \tag{8}$$

where $\boldsymbol{e}$ is white Gaussian noise. To state our results, we need some assessment of the amount of noise, and we will use the following definition of signal-to-noise ratio: $SNR = \|A\|_2^2/\|\boldsymbol{e}\|_2^2$. Moreover, let us consider the following distortion criterion:

$$D_{\ell_2} \quad = \quad \frac{\|\boldsymbol{x}_0 - \hat{\boldsymbol{x}}\|_2}{\|\boldsymbol{x}_0\|_2}, \tag{9}$$

where $\hat{\boldsymbol{x}}$ denotes the estimate. Then, it can be shown that a *sufficient* sampling rate again has the shape

$$\delta \quad \geq \quad O\left(k/n \log(n/k)\right), \tag{10}$$

for all $SNR > 0$ and all $D_{\ell_2} > 0$. The sufficiency of this sampling rate has been shown via polynomial-time algorithms ($\ell_1$-min), see [8], [20]. However, this bound is loose in that little is known about the involved *constants,* and thus, there is no interesting characterization of the trade-offs between the sampling rate, the distortion, and the SNR, other than the following statement: It can be shown that for any $0 < \rho < 1$, there exists a *finite* sampling rate $\delta$ such that

$$D_{\ell_2} \quad = \quad O(1/SNR). \tag{11}$$

A wealth of algorithms have been developed for recovery with respect to an $\ell_2$ criterion (see [6]).

### C. Recovery with small $\ell_0$ distortion

Another naturally arising criterion is the recovery of the sparsity pattern, i.e., the locations of the nonzero elements in the vector $\boldsymbol{x}_0$. We will denote the set of these indices by $\mathcal{S}$. To study this problem, let us consider the same setup as in Section III-B, but restrict attention to the case of *linear* sparsity, i.e., $k = \rho \cdot n$. Additionally define the quantities $P = (1/|\mathcal{S}|)\sum_{i\in\mathcal{S}} x_i^2$ as well as $B = \min_{i\in\mathcal{S}} x_i^2$, leading to the minimum-to-average ratio $MAR = B/P$.[2] Moreover, for simplicity, we will assume that the entries of $A$ are i.i.d. Gaussians.

First, consider the (asymptotic) *exact* recovery of the sparsity pattern $\mathcal{S}$, i.e., the requirement that the probability of exact reconstruction tends to one as $n \to \infty$. For this problem, it was shown in [21] that the necessary sampling rate $\delta$ is *infinite.* Subsequent work [22], [23] has shown that more precisely, the number of required samples is at least $d \geq k+1+\Theta(k \log n)$, which can be attained by a simple thresholding algorithm of complexity linear in $n$.

These negative results say that an excessive number of random projections must be acquired for the task of exact recovery of the sparsity pattern (in the presence of noise), suggesting that this problem is out of reach of the methodology of random projections. Fortunately, it is possible to relax the recovery criterion slightly and obtain a positive result. In fact, for the relaxed problem, sampling requirements are found that closely match those for $\ell_2$ recovery, further supporting random projections as universal signal acquisition.

More precisely, since the degree of sparsity $k = \rho n$ is assumed to be known, the estimated support $\hat{\mathcal{S}}$ has exactly $k$ elements, and we define

$$D_{\ell_0} \quad = \quad 1 - \frac{|\hat{\mathcal{S}} \cap \mathcal{S}|}{k}, \tag{12}$$

which can be interpreted as the percentage of nonzero locations in $\boldsymbol{x}_0$ that were incorrectly recovered. It was shown that for any $0 < SNR < \infty, 0 < MAR \leq 1$, and $0 < D_{\ell_0} < 1$, a *finite* sampling rate $\delta$ is sufficient via the analysis of an exhaustive procedure [21], [24]:

$$\hat{\mathcal{S}}_{\ell_0} = \arg\min_{S} \inf_{\mathbf{u}\in\mathbb{R}^k} \|\boldsymbol{y} - A_S\mathbf{u}\|_2, \tag{13}$$

which can be shown to be equivalent to constrained $\ell_0$-min (for correctly chosen constraints). More recently, it was also shown that even for a simple *thresholding* algorithm given by

$$\hat{\mathcal{S}}_{\text{MC}} = \arg\max_{S} \|A_S^T\boldsymbol{y}\|_2, \tag{14}$$

the sampling rate is still finite [25]. Note that this algorithm merely amounts to sorting the magnitudes of $A^T\boldsymbol{y} \in \mathbb{R}^n$, and is thus of linear complexity in $n$.

Remarkably, by contrast to the problem of recovery within an $\ell_2$ distortion requirement, for approximate sparsity pattern recovery, a set of quite sharp bounds on the sampling rates are available [26]. Together, they establish that the required number of random projections is of a similar behavior as the one for $\ell_2$ recovery. To conclude this section, we give a few illustrations of this. For example, it can be shown that for any $0 < \rho < 1$, there exists a *finite* sampling rate $\delta$ such that

$$D_{\ell_0} \quad = \quad O(1/SNR), \tag{15}$$

by analogy to the result quoted above for $\ell_2$ distortion. More interestingly, the dependence of the sampling rate $\rho$ on the SNR can be characterized as

$$\delta \quad = \quad \rho + \Theta\left(\frac{1}{\log(1+SNR)}\right), \tag{16}$$

for $D_{\ell_0} \ll 1$ and $MAR \ll 1$. A more precise evaluation of the bounds is given in Figure 2 for fixed MAR and $D_{\ell_0}$, illustrating the sharpness of the existing bounds.

## IV. Exploiting Joint Sparsity Among Multiple Sensor Observations

In this section, the discussion will move on from individual sensors at the edge of the network to the base station, which receives multiple sensor observations $\boldsymbol{y}$ from a communication channel. Suppose certain event of interest occurs within the network, then it can be measured by one or more sensors. Clearly in the former case, if a sparse representation exists, the network does not gain any more information to improve the performance. We are more interested in the latter case. More specifically, we will show in several exemplary applications that modeling possible *joint sparsity* shared between multiple

[2] $|\mathcal{S}|$ denotes the cardinality of $\mathcal{S}$.

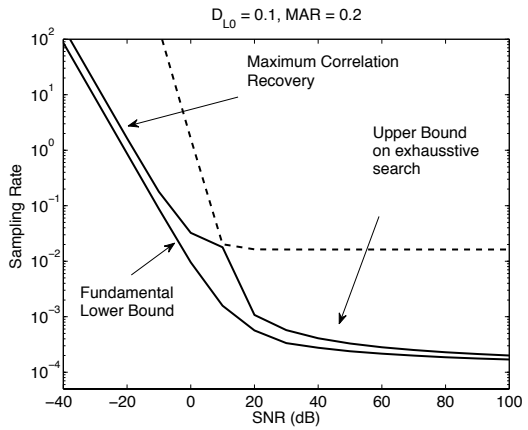


Fig. 2. Sampling requirements for approximate sparsity pattern recovery as a function of SNR. "Exhaustive search" refers to the estimator in (13), and "maximum correlation recovery" to (14). The corresponding performance bounds, along with the fundamental lower bound, are given in [25].

sensor observations is crucial in applying the theory of CS on distributed sensor data.

### A. Distributed Sparsity-Based Classification

We first present a direct application of CS to simultaneous event detection and classification in sensor networks, where individual sensor nodes have sufficient computational capacity and memory to process high-dimensional sensor data. In such configurations, *distributed pattern recognition* becomes possible, where each sensor node is capable of certain decision-making, including classification based on local observations. Only when the local classifier detects a possible occurrence of an event does the sensor node become *active* and transmit the data to the base station. On the base station, a global classifier receives the data from possibly multiple sensor nodes, and further optimizes upon the classification given the local sensor decisions.

A distributed recognition system presents certain unique advantages for sensor network applications. First, good decisions about the validity of the local measurement can reduce the communication between the nodes and the server, and hence reduce the power consumption. Second, although the recognition on the individual sensor nodes is clearly limited by the accuracy of the local observation, such abilities make the design of the global classifier at the network station more flexible. Finally, the ability for individual sensor nodes to make local decisions can be used as feedback to support certain level of autonomous actions without the intervention of a central system.

As an example, we will examine the problem of wearable human action recognition [27], where a network of wearable motion sensors are utilized to recognize certain body actions, such as sitting, running, and going upstairs/downstairs. Figure 4 illustrates some action sequences measured in our experiment. The testbed consists of up to five wearable motion sensors instrumented at different body locations, each of which carries a triaxial accelerometer and a biaxial gyroscope at a sampling rate of 30 Hz. The goal is to detect the temporal support of the actions and correctly classify the actions against a list of possible action categories.

The proposed solution is based on a new classification framework, primarily developed for the classical problem of face recognition [28]. In this framework, the distribution of multiple event classes is modeled as a mixture subspace model, one subspace for each class. Given $C$ classes and a test sample $\boldsymbol{y}$, we seek the sparsest linear representation of the sample with respect to all training examples:

$$\boldsymbol{y} = [A_1, A_2, \cdots, A_C]\boldsymbol{x} + \boldsymbol{e} = A\boldsymbol{x} + \boldsymbol{e}, \tag{17}$$

where the column vectors $\boldsymbol{v}$ of each $A_i$ represent training examples from the $i$th class, and $\boldsymbol{e}$ represents the measurement error. Clearly, if $\boldsymbol{y}$ is a valid test sample, i.e., $\boldsymbol{y}$ is associated with one of the $C$ classes, $\boldsymbol{y}$ can be written as a linear combination of the training samples only from the true class:

$$\boldsymbol{y} = A_i\boldsymbol{x}_i + \boldsymbol{e}. \tag{18}$$

Therefore, the corresponding representation in (17) has a sparse representation $\boldsymbol{x} = [\cdots, \mathbf{0}^T, \boldsymbol{x}_i^T, \mathbf{0}^T, \cdots]^T$: in average only a fraction of $\frac{1}{C}$ coefficients are nonzero, and the dominant nonzero coefficients in sparse representation $\boldsymbol{x}$ reveal the true class.

In order to formulate wearable action recognition in the same classification framework, we first define the notation that we use to describe the distributed sensor data. Suppose in a network of $L$ sensor nodes, each sensor $j$ is capable of measuring $m$-D observations $\boldsymbol{v}^{(j)}$ as stacked accelerometer and/or gyroscope signals over a window of time. For a set of $C$ classes, $n_i$ training examples $A_i^{(j)} \in \mathbb{R}^{m\times n_i}$ shall be collected from the distribution of the $i$-th class on the $j$-th sensor. Now, given a test sample $\boldsymbol{y}^{(j)}$ on sensor $j$, the classification can be easily formulated as solving the following sparse representation:

$$\boldsymbol{y}^{(j)} = [A_1^{(j)}, A_2^{(j)}, \cdots, A_C^{(j)}]\boldsymbol{x} = A^{(j)}\boldsymbol{x} \in \mathbb{R}^m. \tag{19}$$

Equation (19) is the basis to first discuss local classification on the sensor side. Although in theory a sparse solution can be recovered via $\ell_1$-min from (19), in sensor networks, we often need to reduce the dimension of the linear system and thus its complexity. A linear dimensionality reduction function can be defined by choosing a projection $R_j \in \mathbb{R}^{d\times m}$:

$$\bar{\boldsymbol{y}}_j \doteq R_j\boldsymbol{y}_j = R_jA^{(j)}\boldsymbol{x} \doteq \bar{A}^{(j)}\boldsymbol{x} \in \mathbb{R}^d. \tag{20}$$

After projection $R_j$, the feature dimension $d$ typically becomes much smaller than the number $n$ of the training samples: $d \ll n$. Therefore, the new linear system (20) is underdetermined.

In pattern recognition, although $R_j$ can be also viewed as a sensing matrix that essentially reduce the dimensionality of the system (19) as in Section III, the optimality of the projection is rather determined by its *discriminative power*, that is, good dimensionality reduction for classification must preserve the pairwise distance of within-class samples that should be close to each other, and at the same time maximize the sample distances between different classes such that stable decision boundaries can be estimated to partition the distribution of

mixture classes.

Nevertheless, for the classification framework (17) that is based on sparse representation, it was discovered in [28] that if the inherent sparsity is properly sought, the choice of projection $R_j$ is no longer critical. To this end, any Gaussian random matrix performs equally well as many traditional methods such as *principal component analysis* (PCA) and *linear discriminant analysis* (LDA), if sufficient projection dimension is provided. Of course, the disadvantage is also clear: in low-dimensional projection spaces (e.g., $d < 100$), the classification accuracy using random projections would be inferior to those using other discriminative projection methods (e.g., PCA and LDA).

The classification framework (17) also provides an effective means to reject possible invalid observations based on the sparsity assumption. In particular, if a test vector $\boldsymbol{y}^{(j)}$ is not a valid measurement with respect to the $C$ classes, one can show that the dominant coefficients of its sparse representation $\boldsymbol{x}$ should not correspond to any single subspace/class. Then, the notion of *class concentration* of the nonzero coefficients can be used as a threshold to reject invalid outliers [28]. Figure 3 shows a comparison of two $\ell_1$-minimization solutions, one using a valid sample and the other using an outlier.

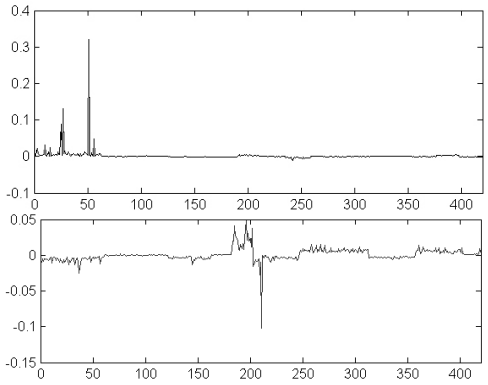

Fig. 3. Top: The dominant coefficients of a valid sample are concentrated in the first action class. Bottom: Coefficients of an outlier are not concentrated in any particular class.

Now, consider at the base station, $L'$ *active* sensors output their measurements ($L' \leq L$). The change in active sensors can be attributed to rejection of invalid samples, sensor failure, or network congestion. Without loss of generality, assume these features are from the first $L'$ sensors: $\bar{\boldsymbol{y}}_1, \bar{\boldsymbol{y}}_2, \cdots, \bar{\boldsymbol{y}}_{L'}$. All the $L'$ measurements, if valid, can only represent one action class. However, short-range sensors such as motion sensors can only make biased decisions based on their own local observations, even if the observations are perfect without noise. For example, a motion sensor located on the upper body could not observe and classify any action of the lower body, and vice versa. It renders the popular majority-voting type mechanism impractical to reach a consistent global decision at the base station. Therefore, we need to construct another layer of global classification to jointly classify the $L'$ samples.

In the work [27], another formulation for global classification of wearable sensors was considered as follows. Denote

$$\bar{\boldsymbol{y}}' = [\bar{\boldsymbol{y}}_1^T, \cdots, \bar{\boldsymbol{y}}_{L'}^T]^T \in \mathbb{R}^{dL'} \tag{21}$$

as the stacked $L'$ sensor features, and the training samples from all the $L$ sensors are collected in the similar fashion:

$$A = [(A^{(1)})^T, (A^{(2)})^T, \cdots, (A^{(L)})^T]^T. \tag{22}$$

Then a global sparse representation $\boldsymbol{x}$ satisfies the following linear system

$$\bar{\boldsymbol{y}}' = \begin{bmatrix} R_1 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & R_{L'} & \cdots & 0 \end{bmatrix} A\boldsymbol{x} = R'A\boldsymbol{x} = \bar{A}'\boldsymbol{x}, \tag{23}$$

where $R'$ is a new projection matrix that only extracts the low-dimensional features from the first $L'$ nodes. Hence, the effect of changing active sensor nodes in the global classification is formulated via the global projection matrix $R'$. The linear system (20) then becomes a special case of (23) where $L' = 1$. The overall algorithm both on the sensors (20) and on the network station (23) is called *distributed sparsity-based classification* (DSC) [27].

Figure 4 demonstrates the results of detection and classification of three human actions using the DSC algorithm. The training samples are manually segmented by human. In the testing step, a sliding window scans through an entire motion sequence along the time axis. False segmentations that correspond to invalid action samples with respect to the training samples are rejected, and the remaining valid samples are classified by the DSC algorithm.

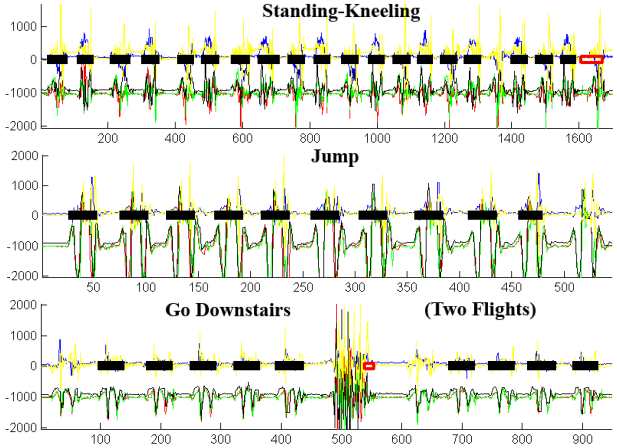


Fig. 4. Detection and classification of three human actions. The plots show readings from the $x$-axis accelerometers over time. The correct classification is indicated as black boxes superimposed in the sequences. The incorrect classification is indicated as red rectangles.

### B. Distributed Compression of Joint Sparse Signals

The previous subsection has presented a distributed classification algorithm (DSC) to classify biased local measurements by short-range motion sensors. Other sensors that measure temperature, light, precipitation, or the electromagnetic field also belong to this category. Another category of sensing modality called long-range sensors is also widely used, including cameras, sonars, and lidars. Long-range sensors typically

consume higher energy than their short-range counterparts. But they also provide much richer information about the environment and dynamic events that take place within the network.

One particular phenomenon that is quite characteristic about a network of long-range sensors is that their fields of view may share a large intersection in 3-D, and hence the environment and the events inside the intersection may be measured by multiple sensors from different vantage points. For example, in object recognition, a common object (or scene) may be observed by multiple surveillance cameras in proximity, and therefore each sensor would obtain a copy of the description of the object. The definition of the object description will be discussed later in the section. Nevertheless, in order to recognize the observed object based on a large object database, which is a computation and memory intensive process, these measurements need to be compressed on the sensor side and transmitted to the base station.

In this subsection, using image-based distributed object recognition as an example [29], we discuss distributed data compression of high-dimensional sensor data when a joint sparse pattern is present. We first define the problem of *distributed compression of joint sparse signals*. Suppose a set of $L$ cameras are equipped to observe a single 3-D object. Each camera $i$ outputs a sparse description of the object $\boldsymbol{x}_i \in \mathbb{R}^n$. Furthermore, the corresponding object images between the $L$ cameras may share a set of common features, which is formulated by the following *joint sparsity* (JS) model [30]:

$$\begin{aligned} \boldsymbol{x}_1 &= \boldsymbol{x}_c + \boldsymbol{z}_1 \in \mathbb{R}^n, \\ &\vdots \\ \boldsymbol{x}_L &= \boldsymbol{x}_c + \boldsymbol{z}_L \in \mathbb{R}^n. \end{aligned} \tag{24}$$

In (24), $\boldsymbol{x}_c$ is called *common sparsity*, and $\boldsymbol{z}_i$ is called *innovation*. Both $\boldsymbol{x}_c$ and $\boldsymbol{z}_i$ are also sparse. Suppose the $L$ cameras communicate with the base station via a band-limited network, and each camera uses a linear encoding function:

$$f_i : \boldsymbol{y}_i = f_i(\boldsymbol{x}_i) \doteq A_i \boldsymbol{x}_i \in \mathbb{R}^{d_i} \quad (d_i < n). \tag{25}$$

Then on the base station, once $\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_L$ are received, we seek simultaneous recovery of the source signals $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_L$.[3]

In computer vision, a sparse representation can be defined to concisely quantize the 2-D appearance of an object in vector form, which is called a SIFT (scale-invariant feature transform) histogram [34], [35]. The definition of SIFT histograms is based on the observation that the object recognition function can be constructed on the basis of decomposing object images into constituent parts (i.e., distinctive image patches). For example, a car figure is comprised of local features such as wheels, windows, car doors, and license plates, etc. Conversely, if these local features are detected from an image, then it implies that one or more cars are present in the image within a neighborhood of the local features. The approach is generally referred to as the *bag-of-words* method [36]. Local features are called *codewords*. Each codeword can be shared among multiple object classes. Hence, the codewords from all object categories can be clustered based on their visual appearance into a *vocabulary* (or codebook). The size of a typical vocabulary ranges from thousands to hundreds of thousands. Given a large vocabulary that contains codewords from many object classes, the histogram representation of a single object image is then *sparse*, as shown in Figures 5 and 6 for two related view points of a toy object.

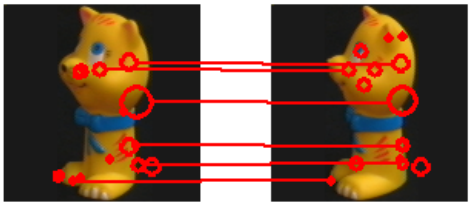

Fig. 5. Detection of interest points (red circles) in two image views of a 3-D toy. The radius of each circle indicates the scale of the interest point in the image. The correspondence of the interest points that are invariant to viewpoint change is highlighted via red lines.

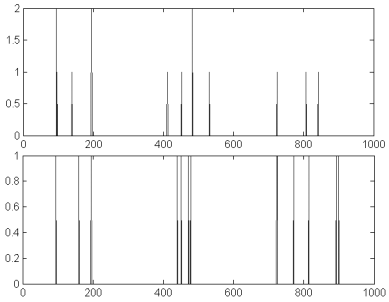


Fig. 6. The histograms representing the image features from the two image views in Figure 5.

In order to simultaneously recover $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_L$, the fact that the common sparsity $\boldsymbol{x}_c$ and innovations $\boldsymbol{z}_i$ in (24) are all sparse leads to the following solution. If we rewrite the random projection on each node based on the JS model as

$$\boldsymbol{y}_i = A_i(\boldsymbol{x}_c + \boldsymbol{z}_i) = A_i \boldsymbol{x}_c + A_i \boldsymbol{z}_i, \tag{26}$$

then an $\ell_1$-min solver can be called to solve the following extended linear system:

$$\begin{aligned} \begin{bmatrix} \boldsymbol{y}_1 \\ \vdots \\ \boldsymbol{y}_L \end{bmatrix} &= \begin{bmatrix} A_1 & A_1 & 0 & \cdots & 0 \\ \vdots & & \ddots & \ddots & \\ A_L & 0 & \cdots & 0 & A_L \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_c \\ \boldsymbol{z}_1 \\ \vdots \\ \boldsymbol{z}_L \end{bmatrix} \\ \Leftrightarrow \quad \boldsymbol{y}' &= A' \boldsymbol{x}' \in \mathbb{R}^D, \quad (D = d_1 + \cdots + d_L). \end{aligned} \tag{27}$$

We note the the most important part $\boldsymbol{x}_c$ in fact indicates the correspondence of object features that are matched across multiple camera views (such as in Figure 5). As the solution to recover it in (27) does not require any assumption about the

[3]Studies of joint sparsity models can be traced back to the problem of *multiple measurement vector* (MMV) [31]–[33]. If all $f_i$ share the same linear projection matrix $A$ and the sparse supports are all the same, then $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_L$ can be simultaneously recovered by solving the following system

$$[\boldsymbol{y}_1, \cdots, \boldsymbol{y}_L] = A[\boldsymbol{x}_1, \cdots, \boldsymbol{x}_L] \Leftrightarrow Y = AX.$$

However, MMV is not suitable for applications such as distributed object recognition because it imposes critical limitations in terms of the distributed signals $\boldsymbol{x}_i$ and the sensing matrices $A_i$. Please refer to [29] for more detail.

relative position between the cameras, nor does it require any prior training information about the appearance of the objects, the distributed encoding method is *viewpoint independent*. In addition, the JS model also improves the sparsity in (27): if the common sparsity $\boldsymbol{x}_c$ dominates the distributed signal, the new coefficient vector $\boldsymbol{x}' = [\boldsymbol{x}_c^T, \boldsymbol{z}_1^T, \cdots, \boldsymbol{z}_L^T]^T$ will have far better sparsity ratio than the individual vectors $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_L$. On the other hand, in the worst case scenario, if no joint sparsity exists, its sparsity ratio is still similar to the average of decoding $L$ projections individually.

Furthermore, taking advantage of the JS model, flexible strategies can be proposed for choosing the random projection dimensions $d_i$. We know in Section III that if each sparse signal $\boldsymbol{x}_i$ is to be decoded independently, the sampling rate should be proportional to the sparsity $k_i \doteq \|\boldsymbol{x}_i\|_0$:

$$\delta_i \doteq \lim_{n\to\infty} d_i/n = O\left(k_i/n \log(n/k_i)\right). \tag{28}$$

With the JS model, a *necessary* condition for simultaneously recovering $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_L$ can be found in [30]. Basically, it requires each sampling rate $\delta_i$ guarantees that the so-called *minimal sparsity signal* of $\boldsymbol{z}_i$ is sufficiently encoded, and the total sampling rate must also guarantee that both the joint sparsity and the innovations are sufficiently encoded.[4] This result suggests a flexible strategy to choose varying sampling rates and communication bandwidth, that is, the random project dimensions $d_i$ need *not* to be the same for the $L$ sensors to guarantee perfect recovery of the distributed data. For example, sensor nodes in a network that have lower bandwidth or lower power reserve can choose to reduce the sampling rate in order to preserve energy.

Figure 7 illustrates how the improved accuracy in distributed data compression translates to better recognition rates [29]. In this experiment, a public object database called COIL-100 was used, which includes multiple-view images of 100 small objects. The SIFT features extracted from the entire image database are quantized to 1000 codewords, i.e., the dimension of the SIFT histograms is 1000. The classifier to match a test histogram vector with training histograms is based on the *support vector machines* (SVMs) method. Figure 7 plots the recognition performance based on several decoding methods:

1) The solid line on the top shows the baseline recognition accuracy assuming no compression is included in the process, and the classifier has direct access to all the SIFT histograms. Hence, the upper-bound *per-view* recognition rate is about 95%.
2) The red curve shows the recognition accuracy directly on the low-dimensional randomly projected feature space $\boldsymbol{y}$.[5] In the low-dimensional regime, classification on random features performs quite well. For example, at 200-D, directly applying SVMs on the random feature space achieves about 88% accuracy.
3) When the dimensions of random projections becomes sufficiently high, the accuracy via $\ell_1$-min overtakes that of the random features, and approaches the baseline performance when the sparse signals are accurately recovered.
4) With more camera views available, enforcing joint sparsity boosts the recognition rate. For example, at 200-D, the average *per-view* recognition rate of a single camera is about 47%, but it jumps to 71% with two camera views, and 80% with three views.

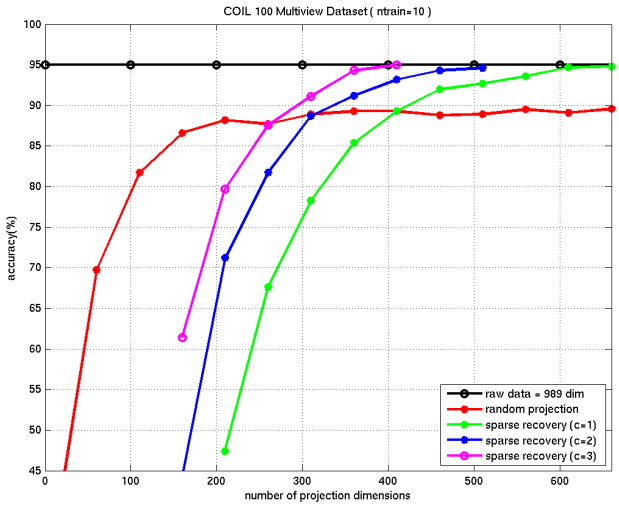


Fig. 7. Per-view classification accuracy versus random projection dimension.

To end this section, we want to point out that distributed object recognition is just one of many applications of the JS model in distributed source coding. Other examples can be found in the literature, including distributed video compression [42], image restoration [43], and analysis of DNA microarrays [44]. The reader can refer to the discussion therein for further reading.

## V. Conclusion and Discussion

We have provided an overview about sparse representation and compressive sensing as a powerful tool to represent and encode high-dimensional signals in the field of sensor networks. The performance metrics for sparsity recovery and inference are primarily based on Gaussian random projections. In some real-world applications, on the other hand, one may be more interested in analyzing specific sensor networks and

[4]The strategy of choosing varying sampling rate is a direct application of the celebrated Slepian-Wolf theorem [37]. In a nutshell, the theorem shows that, given two sources $X_1$ and $X_2$ that generate sequences $x_1$ and $x_2$, asymptotically, the sequences can be jointly recovered with vanishing error probability *if and only if*

$$\begin{aligned} R_1 &> H(X_1|X_2), \\ R_2 &> H(X_2|X_1), \\ R_1+R_2 &> H(X_1,X_2), \end{aligned}$$

where $R$ is the bit rate function, $H(X_i|X_j)$ is the conditional entropy for $X_i$ given $X_j$, and $H(X_i, X_j)$ is the joint entropy [38].

[5]It makes sense to apply classifiers directly on randomly projected subspaces due to another interesting property of random projections. In particular, Johnson-Lindenstrauss lemma [39] shows that, in high-dimensional spaces, Gaussian random projections preserve pairwise $\ell_2$ distance. This result provides another approach to take advantage of random projections without recovering the high-dimensional source signal. Its utility has been demonstrated in WSNs, e.g., feature matching [40] and classification [41].

hence their specific sensing matrices $A$, for instance, sparse sensing matrices. In a resource-constrained situation, one may also be interested in optimizing the columns of $A$ to achieve better sparsity detection and recovery. Existing results in CS theory have provided good solutions to analyze small-sized linear systems, such as the convex polytope theory and the restricted isometry property. For future research, more efficient algorithms are needed to analyze domain specific, medium to large-sized linear systems.

## ACKNOWLEDGMENTS

The authors would like to thank Galen Reeves of the University of California, Berkeley, whose recent research has contributed to part of the paper.

## REFERENCES

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 8, pp. 102–114, 2002.

[2] D. Culler, D. Estrin, and M. Srivastava, "Overview of sensor networks," *Computer*, vol. 8, pp. 41–49, 2004.

[3] P. Baronti, P. Pillai, V. Chook, S. Chessa, A. Gotta, and Y. Hu, "Wireless sensor networks: A survey on the state of the art and the 802.15.4 and zigbee standards," *Computer Communications*, vol. 30, pp. 1655–1695, 2007.

[4] E. Candès, "Compressive sampling," in *Proceedings of the International Congress of Mathematicians*, 2006.

[5] D. Donoho, "For most large underdetermined systems of linear equations the minimal $l_1$-norm solution is also the sparsest solution," *Comm. on Pure and Applied Math*, vol. 59, no. 6, pp. 797–829, 2006.

[6] A. Bruckstein, D. Donoho, and M. Elad, "From sparse solutions of systems of equations to sparse modeling of signals and images," *SIAM Review*, vol. 51, no. 1, pp. 34–81, 2009.

[7] D. Donoho and M. Elad, "Optimally sparse representation in general (nonorthogonal) dictionaries via $\ell^1$ minimization," *PNAS*, vol. 100, no. 5, pp. 2197–2202, 2003.

[8] E. Candès and T. Tao, "Near optimal signal recovery from random projections: Universal encoding strategies?" *IEEE Transactions on Information Theory*, vol. 52, no. 12, pp. 5406–5425, 2006.

[9] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.

[10] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM Review*, vol. 43, no. 1, pp. 129–159, 2001.

[11] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *The Annals of Statistics*, vol. 32, no. 2, pp. 407–499, 2004.

[12] R. Tibshirani, "Regression shrinkage and selection via the LASSO," *Journal of the Royal Statistical Society B*, vol. 58, no. 1, pp. 267–288, 1996.

[13] M. Duarte, M. Davenport, D. Takhar, J. Laska, T. Sun, K. Kelly, and R. Baraniuk, "Single-pixel imaging via compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 83–91, 2008.

[14] D. Donoho and Y. Tsaig, "Fast solution of $\ell^1$-norm minimization problems when the solution may be sparse," *preprint*, 2006.

[15] E. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, 2006.

[16] ——, "Stable signal recovery from incomplete and inaccurate measurements," *Comm. on Pure and Applied Math*, vol. 59, no. 8, pp. 1207–1223, 2006.

[17] L. Gan, T. Do, and T. Tran, "Fast compressive imaging using scrambled block Hadamard ensemble," *preprint*, 2008.

[18] V. Goyal, A. Fletcher, and S. Rangan, "Compressive sampling and lossy compression," *IEEE Signal Processing Magazine*, pp. 48–56, 2008.

[19] D. Donoho and J. Tanner, "Neighborliness of randomly projected simplices in high dimensions," *PNAS*, vol. 102, no. 27, pp. 9452–9457, 2005.

[20] D. Donoho, M. Elad, and V. Temlyakov, "Stable recovery of sparse overcomplete representations in the presence of noise," *IEEE Transactions on Information Theory*, vol. 52, no. 1, pp. 6–18, 2006.

[21] G. Reeves, "Sparse signal sampling using noisy linear projections," M. S. Thesis, UC Berkeley, 2007.

[22] W. Wang, "Information-theoretic limits on sparse signal recovery: Dense versus sparse measurement matrices," in *Proceedings of the International Symposium on Information Theory*, 2008.

[23] A. Fletcher, S. Rangan, and V. Goyal, "Necessary and sufficient conditions on sparsity pattern recovery," *preprint*, 2008.

[24] S. Aeron, M. Zhao, and V. Saligrama, "Sensing capacity of sensor networks: Fundamental tradeoffs of SNR, sparsity and sensing diversity," in *Information Theory and Applications Workshop*, 2007.

[25] G. Reeves and M. Gastpar, "Sampling rates for approximate sparsity recovery," in *Proceedings of the 30th Symposium on Information Theory in the Benelux*, Eindhoven, The Netherlands, 2009.

[26] ——, "Sampling bounds for sparse support recovery in the presence of noise," in *IEEE International Symposium on Information Theory*, 2008.

[27] A. Yang, R. Jafari, S. Sastry, and R. Bajcsy, "Distributed recognition of human actions using wearable motion sensor networks," *Journal of Ambient Intelligence and Smart Environments*, vol. 1, no. 2, pp. 103–115, 2009.

[28] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210 – 227, 2009.

[29] A. Yang, S. Maji, M. Christoudas, T. Darrell, J. Malik, and S. Sastry, "Multiple-view object recognition in band-limited distributed camera networks," in *Proceedings of International Conference on Distributed Smart Cameras*, 2009.

[30] D. Baron, M. Wakin, M. Duarte, S. Sarvotham, and R. Baraniuk, "Distributed compressed sensing," *preprint*, 2005.

[31] B. Rao, "Analysis and extensions of the FOCUSS algorithm," in *The Thirtieth Asilomar Conference on Signals, Systems and Computers*, 1996.

[32] J. Tropp, "Algorithms for simultaneous sparse approximation," *Signal Process*, vol. 86, pp. 572–602, 2006.

[33] Y. Eldar and M. Mishali, "Robust recovery of signals from a structured union of subspaces," *IEEE Transactions on Information Theory*, vol. 55, no. 11, pp. 5302–5316, 2009.

[34] D. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the IEEE International Conference on Computer Vision*, 1999.

[35] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[36] D. Nistér and H. Stewénius, "Scalable recognition with a vocabulary tree," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2006.

[37] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Transactions on Information Theory*, vol. 19, pp. 471–480, 1973.

[38] T. Cover and J. Thomas, *Elements of Information Theory*. Wiley Series in Telecommunications, 1991.

[39] W. Johnson and J. Lindenstrauss, "Extensions of Lipschitz maps into a Hilbert space," *Contemporary Mathematics*, vol. 26, pp. 189–206, 1984.

[40] C. Yeo, P. Ahammad, and K. Ramchandran, "Rate-efficient visual correspondences using random projections," in *Proceedings of the IEEE International Conference on Image Processing*, 2008.

[41] M. Duarte, M. Davenport, M. Wakin, J. Laska, D. Takhar, K. Kelly, and R. Baraniuk, "Multiscale random projections for compressive classification," in *Proceedings of the IEEE International Conference on Image Processing*, 2007.

[42] L. Kang and C. Lu, "Distributed compressive video sensing," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2009.

[43] M. Fornasier and H. Rauhut, "Recovery algorithms for vector valued data with joint sparsity constraints," *SIAM JOurnal on Numerical Analysis*, vol. 46, no. 2, pp. 577–613, 2006.

[44] F. Parvaresh, H. Vikalo, S. Misra, and B. Hassibi, "Recovering sparse signals using sparse measurement matrices in compressed DNA microarrays," *IEEE Journal of Selected Topics in Signal Processing*, vol. 2, no. 3, pp. 275–285, 2008.