Multiple-View Object Recognition in Band-Limited Distributed Camera Networks

Allen Y. Yang, Subhransu Maji, C. Mario Christoudias, Trevor Darrell, Jitendra Malik, and S. Shankar Sastry
Department of EECS, University of California, Berkeley, CA 94720
{yang,smaji,cmch,trevor,malik,sastry}@eecs.berkeley.edu

***Abstract*—In this paper, we study the classical problem of object recognition in low-power, low-bandwidth distributed camera networks. The ability to perform robust object recognition is crucial for applications such as visual surveillance to track and identify objects of interest, and compensate visual nuisances such as occlusion and pose variation between multiple camera views. We propose an effective framework to perform distributed object recognition using a network of smart cameras and a computer as the base station. Due to the limited bandwidth between the cameras and the computer, the method utilizes the available computational power on the smart sensors to locally extract and compress SIFT-type image features to represent individual camera views. In particular, we show that between a network of cameras, high-dimensional SIFT histograms share a joint sparse pattern corresponding to a set of common features in 3-D. Such joint sparse patterns can be explicitly exploited to accurately encode the distributed signal via random projection, which is unsupervised and independent to the sensor modality. On the base station, we study multiple decoding schemes to simultaneously recover the multiple-view object features based on the distributed compressive sensing theory. The system has been implemented on the Berkeley CITRIC smart camera platform. The efficacy of the algorithm is validated through extensive simulation and experiments.**

***Index Terms*—Distributed object recognition, compressive sensing, random projection, joint sparsity, smart camera networks.**

I. Introduction

Object recognition has been a well-studied problem in computer vision. In the traditional formulation, a vision system captures multiple instances of an object from a set of object classes, and is asked to classify a new test image that may contain one or many known object classes. Successful methods have been demonstrated in the past, including pedestrian detection [18], general object detection[1], [29] (e.g., vehicles and animals), and scene annotation [20], [27] (e.g., buildings, highways, and social events). A large body of these works have been based on analysis of certain local image patches that are robust/invariant to image scaling, affine transformation, and visual occlusion, which are the common nuisances in image-based object recognition. The local image patches are typically extracted by a viewpoint-invariant interest point detector [23] combined with a patch descriptor, e.g., SIFT (Scale-Invariant Feature Transform) [21], [4].

In this paper, we consider a relatively new scenario where a network of distributed cameras are set up to simultaneously acquire an ensemble of images when a common object can be viewed from multiple vantage points. In visual surveillance, the ability to *jointly* recognize object classes from multiple views enhances the accuracy of other functionalities such as multiple-view association and tracking. It also effectively compensates visual nuisances in the scene such as occlusion and pose variation. Traditionally, investigators often assume that the cameras are reliably connected to a central computer with no bandwidth limit. On the other hand, the (high-resolution) cameras may not possess significant computational power to locally perform recognition on the object images. As a result, the multiple-view images (or their SIFT representations) would be streamlined back to the computer, and the whole recognition process would be constructed in a centralized fashion at the base station.

Recent studies in distributed object recognition have been mainly focused on two directions. First, when multiple images share a set of common visual features (i.e., affine-invariant interest points), correspondence can be established across camera views. This indeed was the original motivation for the SIFT framework [21]. More recently, [15], [29] proposed to harness the prior spatial distribution of specific features to guide the multiple-view matching process and improve recognition. [8] proposed to utilize SIFT feature matching to obtain a vision graph for an ad-hoc camera network. Taking advantage of random projection, [34] argued that reliable feature correspondence can be estimated in a much lower-dimensional space between cameras communicating under rate constraints.

Second, when the camera sensors do not have sufficient communication resources to streamline the high-dimensional visual features among camera views and perform feature matching, distributed data compression [13], [14] can be utilized to encode and transmit the features. Then the joint object features are recovered at the base station with much higher computational capacity. In particular, [6] proposed a rate-efficient codec to compress scalable tree structures in describing the hierarchy of SIFT histograms. On the other hand, [9] studied a multiple-view SIFT feature selection algorithm. The authors argued that the number of SIFT features that need to be transmitted to the base station can be reduced by considering the joint distribution of the features among multiple camera views of a common object. However, the selection of the joint features depends on learning the mutual information among different camera views, and their relative positions must be fixed.

A. Contributions

We propose a distributed object recognition system suitable for band-limited camera sensor networks. The contributions of this paper are two-fold: First, based on compressive sensing theory, we propose an effective distributed compression scheme to encode SIFT-type object histograms on individual camera sensors. In particular, we explicitly exploit the nonnegativity and the joint sparsity properties in multiple-view histograms to achieve state-of-the-art feature compression for multiple-view recognition. No communication between the cameras is necessary to exchange mutual information about the scene. Random projection will be used to provide dimensionality reduction, which is particularly adept for sensor network applications.

Second, we detail the design of a distributed recognition system. On the sensor side, a smart camera sensor platform called CITRIC [7] is utilized. The substantial computational capability on CITRIC running embedded Linux enables a fast implementation of the SURF (Speeded-Up Robust Features) detector [4] and compression of the object histograms. On the computer side, we demonstrate that the multiple-view object histograms can be jointly recovered with high accuracy using a linear ℓ^1-minimization (ℓ^1-min) solver, called *nonnegative polytope faces pursuit* (PFP). Finally, the object class from the multiple views is classified using *support vector machines* (SVMs). We conduct extensive simulation and a real-world experiment to validate the performance of the system, in which the Columbia COIL-100 object image database [24] is used.

II. Encoding Multiple-View Features via Sparse Representation

Suppose multiple camera sensors are equipped to observe a 3-D scene from multiple vantage points. The sensors communicate with a base-station computer via a single-hop wireless network, i.e., the topology of the network is a star shape with the computer at the center.[1] Using a SURF feature detector, viewpoint-invariant features can be extracted from the corresponding images, as shown in Figure 1. These local features are called *codewords*.

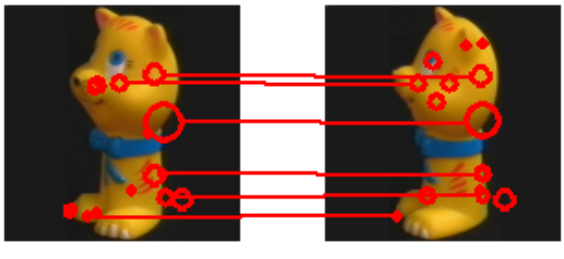

Fig. 1. Detection of interest points (red circles) on two image views of a 3-D toy. The correspondence of the interest points is highlighted via red lines.

[1]The authors have recently proposed a distributed fusion algorithm to compress high-dimensional SIFT histograms in a band-limited multi-hop camera network. The interested reader is referred to [33].

If one is given a large training set of images that capture the appearance of multiple object classes, the codewords from all the object categories then can be clustered based on their visual similarities into a *vocabulary* (or codebook). The clustering normally is based on a hierarchical k-means process [25], [17]. The size of a typical vocabulary ranges from thousands to hundreds of thousands, and naturally each codeword in the vocabulary can be shared among multiple object classes.

Given a large vocabulary that contains codewords from many object classes, the representation of the SIFT features in a single object image is then *sparse*, which is called a SIFT histogram (e.g., a car can be seen to have two to four wheels depending on the viewpoint). Since only a small number of features are exhibited on a specific object, their values (or votes) in the histogram are positive integers, and the majority of the histogram values should be (close to) zero, as shown in Figure 2.[2]

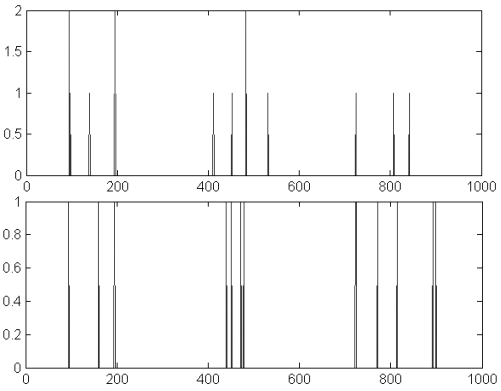

Fig. 2. The histograms representing the image features from the two image views in Figure 1.

We define the problem of multiple-view histogram compression:

Problem 1 (Distributed Compression of Joint Sparse Signals): When L camera sensors are equipped to observe a single 3-D object, the extracted SIFT histograms $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_L \in \mathbb{R}^D$ are assumed to be *nonnegative* and *sparse*. Further, the corresponding images may share a set of common SIFT features from the multiple views:

$$\begin{aligned} \boldsymbol{x}_1 &= \tilde{\boldsymbol{x}} + \boldsymbol{z}_1, \\ &\vdots \\ \boldsymbol{x}_L &= \tilde{\boldsymbol{x}} + \boldsymbol{z}_L. \end{aligned} \tag{1}$$

In (1), $\tilde{\boldsymbol{x}}$ is called the *joint* sparse component, and $\boldsymbol{z}_i$ is called an *innovation* [13]. Both $\tilde{\boldsymbol{x}}$ and $\boldsymbol{z}_i$ are also sparse and nonnegative.

Suppose the L cameras communicate with the base station via a single-hop, band-limited network, and no communication is allowed between the camera pairs:

[2]In this paper, the vocabulary is constructed using a hierarchical k-means algorithm [17], where $k = 10$. For example, a four-level hierarchy generates about 1000 mean values as codewords at the finest level, as shown in the figure.

1) On each camera, construct an encoding function $f : \boldsymbol{x}_i \in \mathbb{R}^D \mapsto \boldsymbol{y}_i \in \mathbb{R}^d$ ($d < D$) that compresses the histogram.
2) On the base station, once $\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_L$ are received, simultaneously recover the histogram signals $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_L$ and classify the object class in 3-D.

A. Random Projection

We first discuss choosing a projection function to encode the histogram vectors $\boldsymbol{x}$ in a lower-dimensional space. In particular, a linear projection function is defined as:

$$f : \boldsymbol{y} = A\boldsymbol{x}, \tag{2}$$

where $A \in \mathbb{R}^{d \times D}$ is in general a full-rank matrix with $d < D$, and represents an overcomplete coding dictionary.

In this paper, we assume that information sharing between cameras is not available for the compression process. This constraint is particularly relevant in distributed camera networks, where the locations of the cameras may not be known and may not be assumed fixed before hand. Recently, a special projection method called *random projection* has gained much publicity in applications where the prior information of the source data and the computational power of the sensor modalities are limited [5], [7], [34]. In this case, each element a_{ij} of A is independently drawn from a zero-mean Gaussian distribution. One can further simplify the implementation for generating random matrices by a Bernoulli distribution of two values $(+1, -1)$ with equal probability, i.e., the Rademacher distribution.

Compared with other linear projections, the main advantages of random projection are two-fold: 1. Random projection is efficient to generate using a pseudo-random number generator, and it does not depend on any domain-specific training set. 2. In terms of robustness to wireless congestion and packet loss, if (part of) the projected coefficients are dropped from the communication, the node needs not resend the coefficients, so long as the receiver can keep track of the packet IDs to reconstruct a partial random matrix with a lower dimension d in (2). In addition, it is straightforward to implement a progressive compression protocol to construct additional random projections of the signal $\boldsymbol{x}$ to improve the reconstruction accuracy.

Clearly, one implication of encoding $\boldsymbol{x}$ in (2) is that the possible solution becomes not unique. In the following, we establish the basic compressive sensing framework to recover $\boldsymbol{x}$ on each individual camera. In Section II-C, we will investigate new methods to recover joint sparse signals from multiple cameras.

Before we discuss how to recover histogram vectors $\boldsymbol{x}$ in (2), remember the goal of Problem 1 is to classify the object class in the 3-D scene. Indeed, one can directly utilize the randomly projected features $\boldsymbol{y}$ in the d-dim feature space for recognition. One important property of random projection is that it preserves the pairwise Euclidean distance, known as the *Johnson-Lindenstrauss* (J-L) lemma [3]:

Theorem 1 (Johnson-Lindenstrauss Lemma): Let $0 < \epsilon < 1$ and an integer n for the number of any point cloud $\mathcal{X} \subset \mathbb{R}^D$. For any $d \geq 4(\epsilon^2/2 - \epsilon^3/3)^{-1} \log n$, a random projection $f : \mathbb{R}^D \to \mathbb{R}^d$ preserves the pairwise Euclidean distance with high probability:

$$(1-\epsilon)\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2 \leq \|f(\boldsymbol{x}_i) - f(\boldsymbol{x}_j)\|_2^2 \leq (1+\epsilon)\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2, \tag{3}$$

where $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are any two points in the point cloud $\mathcal{X}$.

The J-L lemma essentially provides the following guarantee: For applications where only pairwise ℓ^2-distances are concerned, it suffices to use the randomly projected features $\boldsymbol{y}$ and "throw away" the original data. In machine learning, random projection has been applied to reducing the data complexity for k-nearest neighbor (kNN) [2], [34]. On the other hand, Gaussian random projection does not guarantee the bounds for other ℓ^p-norms with $(p < 2)$. Particularly in object recognition, the similarity between different histograms is often measured w.r.t. the ℓ^1-norm using the histogram intersection kernel, which will be discussed in more detail in Section III-B. More recent studies have proposed other p-*stable* random projection methods to effectively approximate the general ℓ^p-distance [32], [19], for example, using Cauchy random projections to preserve the pairwise ℓ^1-distance. For clarity, in the paper, our discussion will be limited to Gaussian/Rademacher random projections.

Another observation in the J-L lemma is that the lower bound of the projection dimension d depends on the number of samples n. However, the lemma does not assume any special structure of the point cloud in the high-dimensional space. If we further assume the source signal $\boldsymbol{x}$ is *sufficiently* sparse, e.g., as the case for image feature histograms computed over a large vocabulary, each $\boldsymbol{x}$ then can be reliably recovered/decoded from its random observations $\boldsymbol{y}$. This "inverse" process is related to the *sparsity* of the source signal $\boldsymbol{x}$, which is the main subject in compressive sensing [5], [11].

Theorem 2: Given a sparse signal $\boldsymbol{x}_0$, denote k as the sparsity (i.e., $\|\boldsymbol{x}_0\|_0 = k$). Then for large D, with high probability, there exists a constant $\rho = \rho(A)$ in (2) such that for every $\boldsymbol{x}_0$ with its sparsity $k < \rho d$, $\boldsymbol{x}_0$ is the unique solution of the ℓ^1-min program:

$$(P_1): \quad \min \|\boldsymbol{x}\|_1 \text{ subject to } \boldsymbol{y} = A\boldsymbol{x}. \tag{4}$$

Clearly, the condition ρ in Theorem 2 is a function of the matrix A. In fact, for a particular A matrix, ρ can be exactly quantified in convex polytope theory [10], [12]. This relationship is also pivotal in enforcing the nonnegativity of $\boldsymbol{x}$ that will be discussed in Section II-C. In the rest of this subsection, we will first overview this relationship.

Figure 3 illustrates a projection between a cross polytope $C \doteq C^3 \subset \mathbb{R}^3$ and its image $AC \subset \mathbb{R}^2$. In general, a cross polytope C^D in $\mathbb{R}^D$ is the collection of vectors $\{x : \|\boldsymbol{x}\|_1 \leq 1\}$. For any k-sparse vector $\boldsymbol{x}$, $\|\boldsymbol{x}\|_1 = 1$, one can show that $\boldsymbol{x}$ must lie on a $(k-1)$-face of C^D. With projection $A \in \mathbb{R}^{d \times D}$, AC is an induced *quotient polytope* in the d-dim space. It is

important to note that some of the vertices and faces of C may be mapped to the interior of AC, i.e., they do not "survive" the projection.

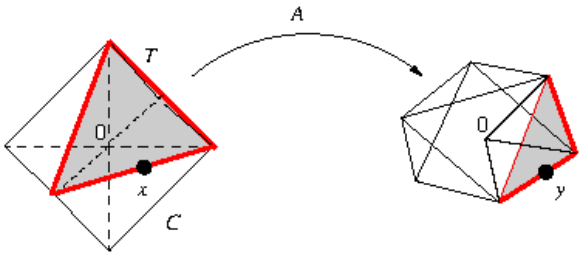

Fig. 3. Projection of a cross polytope C in $\mathbb{R}^3$ to a quotient polytope AC via projection A. The corresponding simplex is T at the shaded area. Both AC and AT are 0-neighborly.

Theorem 3: 1) For a projection matrix $A \in \mathbb{R}^{d\times D}$, the quotient polytope AC is called k**-neighborly** if all the k-faces of C^D are mapped to the boundary of AC. Any sparse signal $\boldsymbol{x} \in \mathbb{R}^D$ with $(k+1)$ or less sparse coefficients can be recovered by (P_1) if and only if AC is k-neighborly.

2) For a specific $(k+1)$-sparse signal $\boldsymbol{x} \in \mathbb{R}^D$, $\boldsymbol{x}$ must lie on a unique k-face $F \subset C$. Then $\boldsymbol{x}$ can be uniquely recovered by (P_1) if and only if AF is also a k-face of AC.

Theorem 3 is a powerful tool to examine if a sparse signal under a projection A can be uniquely recovered by (P_1). For example, in Figure 3, AC is 0-neighborly. Therefore, any 1-sparse signal can be uniquely recovered by (P_1). However, for a specific $\boldsymbol{x}$ on a 1-face of C, $\boldsymbol{x}$ is 2-sparse and it is projected to a 1-face of AC. Hence, $\boldsymbol{x}$ also can be uniquely recovered via (P_1).

For a specific A matrix that depends on the application, one can simulate the projection by sampling vectors $\boldsymbol{x}$ on all the k-faces of C. If with high probability, the projection $A\boldsymbol{x}$ survives (i.e., on the boundary of AC), then AC is at least k-neighborly. The simulation provides a practical means to verify the neighborliness of a linear projection, particularly in high-dimensional data spaces. On the other hand, a somewhat surprising property guarantees the well-behavior of random projection: In a high-dimensional space, with high probability, random projection preserves most faces of a cross polytope. A short explanation to this observation is that most randomly generated column vectors in A are linearly independent.

B. Enforcing Nonnegativity using Polytope Faces Pursuit

Given the observations $\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_L$, (P_1) in the previous subsection provides a solution to *independently* recover each of the ensemble elements $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_L$. However, such a solution fails to observe that in our application the sparse signals $\boldsymbol{x}$ represent image histograms and are therefore strictly nonnegative, and it also fails to enforce the possible joint sparse pattern that is shared among multiple camera views. In this subsection, we show that the abilities for a set of new algorithms to enforce nonnegativity and joint sparsity of the ensemble significantly boost the accuracy of ℓ^1-min. In Section IV, we will further demonstrate that the improvement also leads to better classification of 3-D objects.

We first discuss how to impose nonnegativity in general ℓ^1-min. Assuming nonnegative $\boldsymbol{x}$ is normalized to be ℓ^1-norm one without loss of generality, we denote $T \doteq T^{D-1}$ as the standard simplex in $\mathbb{R}^D$, i.e.,

$$T = \{\boldsymbol{x} : \|\boldsymbol{x}\|_1 = 1 \text{ and } \boldsymbol{x} \geq 0\}. \tag{5}$$

Figure 3 shows the relationship between C^D and T^{D-1}. Hence, the nonnegative vector $\boldsymbol{x}$ must lie on a $(k+1)$-face of T, which is a small subset of the cross polytope. The following theorem shows that the nonnegativity constraint reduces the domain of possible solutions for ℓ^1-min (as shown in Figure 3) [12]:

Theorem 4: 1) Any nonnegative sparse signal $\boldsymbol{x} \in \mathbb{R}^D$ with $(k+1)$ or less sparse coefficients can be recovered by

$$(P_1'): \quad \min \|\boldsymbol{x}\|_1 \text{ subject to } \boldsymbol{y} = A\boldsymbol{x} \text{ and } \boldsymbol{x} \geq 0 \tag{6}$$

if and only if all k-faces of T^{D-1} survive the projection A.

2) For a specific nonnegative $(k+1)$-sparse signal $\boldsymbol{x}$, $\boldsymbol{x}$ must lie on a unique k-face $F \subset T$. Then $\boldsymbol{x}$ can be uniquely recovered by (P_1') if and only if AF is also a k-face of AT.

The nonnegative ℓ^1-min (6) is a linear program, and can be solved by many algorithms, including *orthogonal matching pursuit* (OMP), *basis pursuit* (BP), and *polytope faces pursuit* (PFP) [26]. These algorithms are usually preferred in sensor network applications compared to other more expensive quadratic programs (e.g., the LASSO [30]). In both simulation and experiments on real-world data, we have found that PFP is a more efficient algorithm than the LASSO to impose the nonnegativity constraint, and produces good results that are difficult to solve for OMP.

The PFP algorithm is summarized in Algorithm 1. Note that the modification of Algorithm 1 compared to the standard PFP that permits negative coefficients is that the pursuit does not involve the antipodal vertices as the columns of $-A$. In the rest of the paper, Algorithm 1 is the ℓ^1-solver of our choice.

C. Estimation of Joint Nonnegative Sparse Signals

In this subsection, we propose a novel solution to simultaneously recover the ensemble $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_L$ that represent L camera views, where a joint sparse pattern $\tilde{\boldsymbol{x}}$ in (1) may be present.

A straightforward attempt to recover joint sparsity formulates the ensemble of randomly projected vectors as *multiple measurement vectors* (MMV) [28], [31]:

$$[\boldsymbol{y}_1, \cdots, \boldsymbol{y}_L] = A[\boldsymbol{x}_1, \cdots, \boldsymbol{x}_L] \Leftrightarrow Y = AX. \tag{7}$$

Assuming the ensembles $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_L$ are sparse and share the same support, one can modify (P_1') as

$$(P_{2,1}'): \min \sum_{i=1}^{D} \|\boldsymbol{x}^{(i)}\|_2 \text{ subject to } Y = AX, \boldsymbol{x}^{(i)} \geq 0, \tag{8}$$

Algorithm 1 Nonnegative Polytope Faces Pursuit (PFP)

Input: A full rank matrix $A = [\boldsymbol{a}_1, \cdots, \boldsymbol{a}_D] \in \mathbb{R}^{d\times D}$, $d < D$, a vector $\boldsymbol{y} \in \mathbb{R}^d$, and an error threshold ϵ.

1: Initialization: $k \leftarrow 0$. Assign residual $\boldsymbol{r}_0 \leftarrow \boldsymbol{y}$, sparse support index set $\Omega \leftarrow \emptyset$, $\boldsymbol{x} \leftarrow 0 \in \mathbb{R}^D$.
2: Project $\boldsymbol{x}$ onto the boundary of the dual polytope of A: $\boldsymbol{c}_0 \leftarrow 0$.
3: **repeat**
4: $k \leftarrow k + 1$.
5: Pursuit on the dual polytope faces

$$i = \arg\min_{j\notin\Omega}\{\alpha | \boldsymbol{a}_j^T(\boldsymbol{c}_{k-1} + \alpha \boldsymbol{r}_{k-1}) = 1\},$$

$\Omega \leftarrow \Omega \cup \{i\}$.
6: Update: $\boldsymbol{x}^\Omega \leftarrow (A^\Omega)^\dagger \boldsymbol{y}$, $\boldsymbol{r}_k = \boldsymbol{y} - A\boldsymbol{x}$.
7: Project on the dual polytope: $\boldsymbol{c}_k = ((A^\Omega)^\dagger)^T \mathbf{1}$.
8: **if** $\boldsymbol{x}$ contains negative coefficients **then**
9: Remove negative support indices from Ω.
10: Go to STEP 6.
11: **end if**
12: **until** $\|\boldsymbol{r}_k\|_2 < \epsilon$.

Output: $\boldsymbol{x}$.

where $\boldsymbol{x}^{(i)}$ is the ith row of X. By a similar argument in compressive sensing, one can show that if $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_L$ are sufficiently sparse and share the same support (nonzero rows), they can be uniquely recovered by $(P'_{2,1})$.

However, the drawbacks of the MMV formulation are also obvious in the context of distributed object recognition: First, MMV imposes that during the encoding process, all camera sensors must share the same projection matrix A. It is clearly more desirable for individual sensors to have the freedom to construct their own projections. Second, MMV does not take into account any possible innovations $\boldsymbol{z}_i$ in different camera views in (1). As shown in Figure 2, the assumption is ill-posed as innovation features abound in multiple-view SIFT features.

In this paper, we propose a *sparse innovation model* (SIM) to directly recover the joint sparse signal and the sparse innovations as the following:

$$\begin{aligned} \boldsymbol{y}_1 &= A_1(\tilde{\boldsymbol{x}} + \boldsymbol{z}_1) = A_1\tilde{\boldsymbol{x}} + A_1\boldsymbol{z}_1, \\ &\vdots \\ \boldsymbol{y}_L &= A_L(\tilde{\boldsymbol{x}} + \boldsymbol{z}_L) = A_L\tilde{\boldsymbol{x}} + A_L\boldsymbol{z}_L, \end{aligned} \tag{9}$$

where both $\tilde{\boldsymbol{x}}$ and the ensemble of $\boldsymbol{z}_1, \cdots, \boldsymbol{z}_L$ are assumed to be nonnegative. The SIM can be directly solved in the following linear system via PFP:

$$\begin{aligned} \begin{bmatrix} \boldsymbol{y}_1 \\ \vdots \\ \boldsymbol{y}_L \end{bmatrix} &= \begin{bmatrix} A_1 & A_1 & 0 & \cdots & 0 \\ \vdots & & \ddots & \ddots & \\ A_L & 0 & \cdots & 0 & A_L \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{x}} \\ \boldsymbol{z}_1 \\ \vdots \\ \boldsymbol{z}_L \end{bmatrix} \\ \Leftrightarrow \quad \boldsymbol{y}' &= A'\boldsymbol{x}' \in \mathbb{R}^{dL}. \end{aligned} \tag{10}$$

The global projection function (10) projects a $D(L+1)$-dim nonnegative sparse signal onto a dL-dim subspace defined by matrix A'. The new linear system also improves the sparsity w.r.t. the total data space. As an example, suppose for each camera in (2), $\rho = \frac{k}{d}$, and $\|\tilde{\boldsymbol{x}}\|_0 = \frac{k}{2}$ and $\|\boldsymbol{z}_i\|_0 = \frac{k}{2}$. Then the new sparsity ratio in (10) becomes

$$\rho' = \frac{(L+1)k/2}{dL} = \frac{L+1}{2L}\rho. \tag{11}$$

Hence, with a large L for the number of the cameras, the joint histogram $\boldsymbol{x}'$ becomes much sparser and can be recovered more accurately via ℓ^1-min.

Example 1: We validate the performance of the three joint sparsity solutions using sythetic data. Experiments on real-world multiple-view image data will be presented in Section IV. Suppose the triplet $D = 1000$, $d = 200$, and $k = 60$. To simulate multiple-view histograms, three k-sparse histograms $\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3 \in \mathbb{R}^{1000}$ are randomly generated with nonzero coefficients between 0 and 1, and then randomly projected to a 200-dim space. Among the 60 nonzero coefficients in each histogram, different combinations of joint sparsity and innovation are constructed, as shown in Table I, from $\|\tilde{\boldsymbol{x}}\|_0 = 60$ and $\|\boldsymbol{z}\|_0 = 0$ to $\|\tilde{\boldsymbol{x}}\|_0 = 30$ and $\|\boldsymbol{z}\|_0 = 30$. We evaluate the performance of OMP, PFP, MMV, and SIM, based on their ℓ^0-norm distortion (i.e, sparse support error) and ℓ^2-norm distortion between the ground truth and the estimate.

TABLE I
AVERAGE ℓ^0-ERROR AND ℓ^2-ERROR OF OMP, PFP, MMV, AND SIM OVER 100 TRIALS. THE TWO NUMBERS IN THE PARENTHESES INDICATE THE SPARSITY IN THE JOINT SPARSE SIGNAL AND THE INNOVATION, RESPECTIVELY. THE BEST RESULTS ARE INDICATED IN BOLD NUMBERS.

Sparsity	(60,0)	(40,20)	(30,30)
ℓ^0_{OMP}	56.14	56.14	56.14
ℓ^2_{OMP}	1.76	1.76	1.76
ℓ^0_{PFP}	3.48	3.48	3.48
ℓ^2_{PFP}	0.05	0.05	0.05
ℓ^0_{MMV}	42.17	48.77	58.50
ℓ^2_{MMV}	1.84	3.10	3.67
ℓ^0_{SIM}	**1.85**	**1.65**	**1.95**
ℓ^2_{SIM}	**0.02**	**0.02**	**0.02**

First, since both OMP and PFP do not consider any joint sparsity, each $\boldsymbol{x}$ is independently recovered from its projection $\boldsymbol{y}$. Hence, their performance should not change w.r.t. different sparsity combinations. Without enforcing the nonnegativity, OMP basically fails to recover any meaningful nonnegative sparse patterns. On the other hand, the average sparse support error for PFP that enforces the nonnegativity is much smaller.

When the joint sparsity is considered, the model of MMV does not perform well either. It clearly underperforms both individual PFP and the SIM, even in the case of (60,0) without innovation signals.

Overall, the SIM achieves the best performance. First, w.r.t. different combinations of joint sparsity and innovation, the average support error stays consistent, which shows the method adapts well to the presence of innovation signals in the multiple-view histograms. More importantly, the method achieves a very low estimation error both in ℓ^0 and ℓ^2. Out of 60 nonzero coefficients, only one coefficient is misidentified.

III. System Implementation

A. Feature Extraction on CITRIC Camera Motes

The complete recognition system has been implemented on the Berkeley CITRIC smart camera sensor [7] and a computer as the base station. The design of the CITRIC platform provides considerable computational power to execute SIFT feature extraction and histogram compression on the sensor board. Each CITRIC mote consists of a camera sensor board running embedded Linux and a TelosB network board running TinyOS. The camera board integrates a 1.3 megapixel SXGA CMOS image sensor, a frequency-scalable (up to 624 MHz) microprocessor, and up to 80 MB memory. We have ported an Open SURF library to extract SIFT features.[3] Figure 4 illustrates two examples.

Fig. 4. The interest points detected from a corridor scene (left) and a tree object (right). The SURF features are superimposed as red circles.

The TelosB network board uses the IEEE 802.15.4 protocol to communicate between camera nodes and the base station. The typical bandwidth is 250 Kbps. To measure the speed of the system on the camera sensor, we have conducted a real-world experiment at multiple locations of an office building [33]. Overall, the CITRIC system takes about 10–20 seconds to extract SURF features from 320×240 grayscale images and transmits the compressed histograms $\boldsymbol{y}$ to the base station, depending on the number of SURF features and the dimension of the random projection. The experiment reveals some limitation of the CITRIC platform in more computation-intensive applications such as real-time SIFT feature extraction. We believe the limitation can be mitigated in a future hardware update with a state-of-the-art floating-point mobile processor and a faster data rate between the CITRIC mote and the network mote.

B. Multiple-View Recognition via SVMs

On the base station, upon receiving the compressed features from the L cameras, the original sparse histograms $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_L$ are simultaneously recovered via the SIM (10). In order to identify M object classes w.r.t. each individual camera view, we train one-vs-one SVM classifiers for every pair of categories. We use LibSVM[4] with the histogram intersection kernel for learning classifiers. This kernel and its variants such as the pyramid match kernel [16] have been shown to work quite well for visual recognition. Since there have been quite efficient algorithms for classification [22], we do not believe the possible computation and memory limitation would be a major issue for a real-time classification system on the base station computer.

When multiple views are available there are various ways to use them together to improve recognition. The simplest being one that enforces agreement between the views by means of majority voting. One may also learn to classify in the joint representation domain directly. On the other hand, most existing methods must assume the relative camera positions are known and fixed. We leave the exploration of this direction for future research.

IV. Experiment

To demonstrate the performance of the algorithm on real multiple-view images, we utilize the public COIL-100 dataset. This dataset consists of 72 views of 100 objects imaged from 0 to 360 degrees in 5 degree increments. In this setting we perform instance-level recognition and demonstrate the performance of our approach with varying number of random projection dimensions. The imaging process on the CITRIC mote is simulated by directly uploading the COIL images to the camera memory for processing.

A local feature representation was computed for each image using 10-D PCA-SIFT features extracted on a regular grid with a 4 pixel spacing that were combined with their image location to form a 12-D feature space. The features from a subset of the COIL-100 images were then used to compute the vocabulary of a multi-resolution histogram image representation found using hierarchical k-means with LIBPMK [17]. We used 4 levels and a branching factor of 10 to give a 991 word vocabulary at the finest level of the hierarchy, as shown in Figure 5. We represent each image and perform L1 recovery using the finest level of the hierarchical histogram, and similarity between images is computed using histogram intersection over the resulting 991-D histogram vectors corresponding to each image. Note that it is straightforward to reconstruct the higher levels of the hierarchy given the reconstructed image histogram at the finest level, and other metrics such as pyramid match similarity can also be used.

In this paper, 10 training examples are sampled uniformly from the complete 360 degree viewing circle. Given a new test example that is not in the training set, each pairwise SVM classifier votes for a class of the example, and the final decision is assigned to the class with the maximum number of votes. For testing using multiple views we use the projected features from the neighboring views of the test example in the dataset

[3]The Open SURF project is documented at: http://code.google.com/p/opensurf1/.

[4]http://www.csie.ntu.edu.tw/~cjlin/libsvm/.

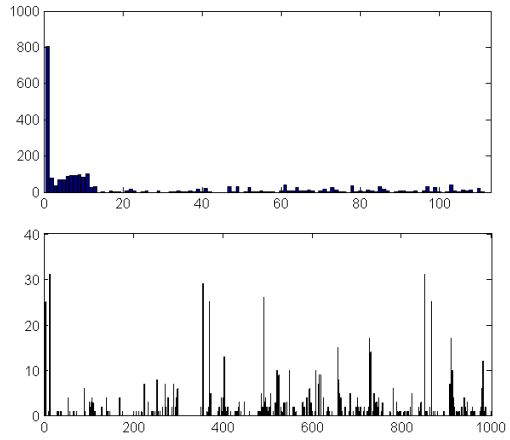

Fig. 5. Example of a hierarchical histogram tree. Top: Level 1–3. Bottom: Level 4.

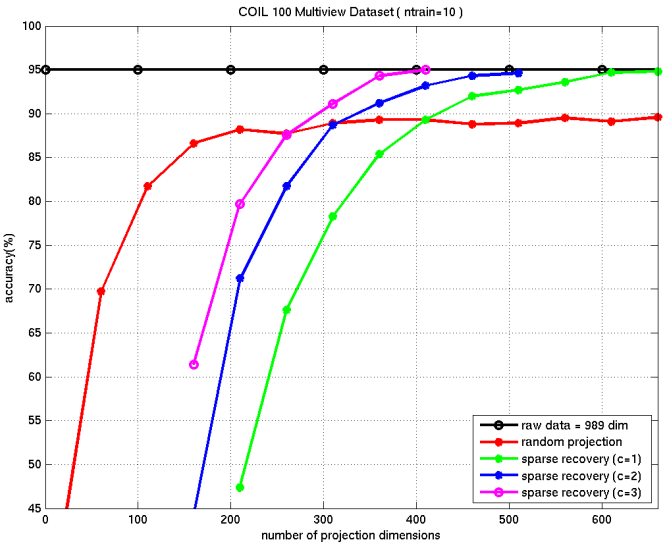

Fig. 6. Per-view classification accuracy (in color) w.r.t. random projection dimensions. Three compression schemes are tested: 1. Randomly projected feature space. 2. PFP on each independent view. 3. SIM with 2-to-3 camera views. The top curve is the baseline recognition rate without histogram compression, and it is independent to the dimension change.

to jointly recover the features, but classify each independently for a fair comparison.

Figure 6 shows the performance of various methods on this dataset. The solid line on the top shows the ground-truth recognition accuracy assuming no compression is included in the process, and the computer has direct access to all the SIFT histograms. Hence, the upper-bound per-view recognition rate is about 95%. When the histograms are compressed via random projection, in the low-dimension regime, the random projection space works quite well to directly classify the object classes. For example, at 200-D, directly applying SVMs in the random projection space achieves about 88% accuracy. However, the accuracy soon flattens out and is overtaken by the ℓ^1-min methods when the projected feature dimension becomes high enough.

Since the ℓ^1-min scheme provides a means to recover the original SIFT histogram in the high-dimensional space, when the dimension of random projection becomes sufficiently high, the accuracy via PFP surpasses the random projection features, and approaches the baseline performance beyond 600-D. Furthermore, when more camera views are available, the joint sparsity model of SIM significantly boosts the accuracy by as much as 50%, as seen in Figure 6. For example, at 200-D, the recognition accuracy for PFP is about 47%, but it jumps to 71% with two camera views, and 80% with three views. SIM has also been shown to reduce the ℓ^1- and ℓ^2-recovery-error in Figure 7.

V. Conclusion and Discussion

We have studied the problem of distributed object recognition in band-limited smart camera networks. The main contribution of the solution is a novel compression framework that encodes SIFT-based object histograms. We exploit three important properties of multiple-view image histograms of a 3-D object: histogram sparsity, histogram nonnegativity, and multiple-view joint sparsity.

Inspired by compressive sensing theory, Gaussian random projection has been proposed as a universal dimensionality reduction function to compress high-dimensional histograms. We have discussed the implication of the J-L lemma to random projection that classification can be directly applied to the randomly projected samples in the low-dimensional space as the projection preserves the pairwise Euclidean distance.

The disadvantage of the randomly projected samples is that their pairwise ℓ^1-distance is not preserved, which is crucial for generating accurate object recognition results based on SIFT feature histograms. We have proposed a sparse innovation model to directly characterize the relationship between the joint sparsity and the innovation signals in multiple views. We have further shown that both the joint sparsity and the innovations are sparse and nonnegative. Hence, they can be simultaneously recovered on the base computer via a ℓ^1-min solver such as the nonnegative polytope faces pursuit. The complete recognition system has been implemented on the Berkeley CITRIC smart camera platform.

One of the limitations in the current solution is that the algorithm lacks a mechanism to classify and associate multiple objects in the scene. This is due to the fact that each histogram is being treated as a holistic representation of the 3-D scene. Another limitation is that the classification via SVMs is conducted on a per-view basis, although majority-voting can be trivially applied to incorporate the multiple views to some extent. Future solutions to these questions must carefully study the detailed structure of sparse histograms in full granularity, and answer how the association of these SIFT features can improve the classification across multiple camera views in a band-limited sensor network.

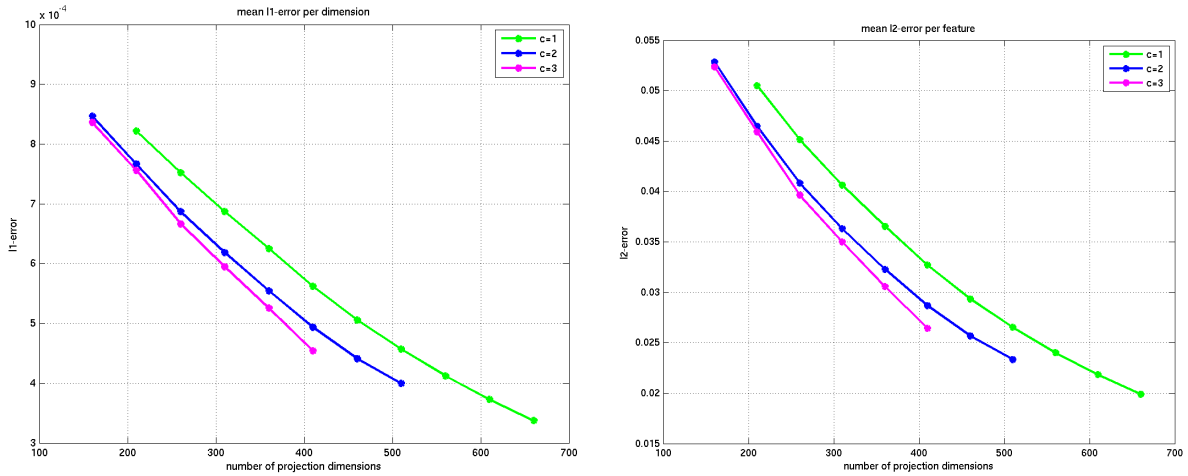

Fig. 7. **(Left)** ℓ^1-error per dimension and **(Right)** ℓ^2-error per features for the recovered features using $1, 2$ and 3 cameras.

ACKNOWLEDGMENTS

This work was supported in part by ARO MURI W911NF-06-1-0076. The authors thank Kirak Hong and Posu Yan of the University of California, Berkeley, for the implementation of the SURF function on the Berkeley CITRIC camera platform.

REFERENCES

[1] S. Agarwal and D. Roth. Learning a sparse representation for object detection. In *ECCV*, 2002.

[2] N. Ailon and B. Chazelle. Approximate nearest neighbors and the fast Johnson-Lindenstrauss transform. In *STOC*, 2006.

[3] R. Baraniuk, M. Davenport, R. de Vore, and M. Wakin. The Johnson-Lindenstrauss lemma meets compressed sensing. *to appear in Constructive Approximation*, 2007.

[4] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. *CVIU*, 110(3):346–359, 2008.

[5] E. Candès and T. Tao. Near optimal signal recovery from random projections: Universal encoding strategies? *IEEE Trans. Information Theory*, 52(12):5406–5425, 2006.

[6] D. Chen, S. Tsai, V. Chandrasekhar, G. Takacs, J. Singh, and B. Girod. Tree histogram coding for mobile image matching. In *Data Compression Conference*, 2009.

[7] P. Chen, P. Ahammad, C. Boyer, S. Huang, L. Lin, E. Lobaton, M. Meingast, S. Oh, S. Wang, P. Yan, A. Yang, C. Yeo, L. Chang, D. Tygar, and S. Sastry. CITRIC: A low-bandwidth wireless camera network platform. In *Proceedings of the International Conference on Distributed Smart Cameras*, 2008.

[8] Z. Cheng, D. Devarajan, and R. Radke. Determining vision graphs for distributed camera networks using feature digests. *EURASIP Journal on Advances in Signal Processing*, pages 1–11, 2007.

[9] C. Christoudias, R. Urtasun, and T. Darrell. Unsupervised feature selection via distributed coding for multi-view object recognition. In *CVPR*, 2008.

[10] D. Donoho. Neighborly polytopes and sparse solution of underdetermined linear equations. *preprint*, 2005.

[11] 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*, 59(6):797–829, 2006.

[12] D. Donoho and J. Tanner. Neighborliness of randomly projected simplices in high dimensions. *PNAS*, 102(27):9452–9457, 2005.

[13] M. Duarte, S. Sarvotham, D. Baron, M. Wakin, and R. Baraniuk. Distributed compressed sensing of jointly sparse signals. In *the Thirty-Ninth Asilomar Conference on Signals, Systems and Computers*, 2005.

[14] Y. Eldar and M. Mishali. Robust recovery of signals from a union of subspaces. Technical report, (preprint) arXiv:0807.4581, 2008.

[15] V. Ferrari, T. Tuytelaars, and L. Van Gool. Integrating multiple model views for object recognition. In *CVPR*, 2004.

[16] K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *ICCV*, pages 1458–1465, 2005.

[17] J. Lee. Libpmk: A pyramid match toolkit. Technical Report MIT-CSAIL-TR-2008-017, MIT CSAIL, 2008.

[18] B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In *CVPR*, pages 878– 885, 2005.

[19] P. Li, T. Hastie, and K. Church. Nonlinear estimators and tail bounds for dimension reduction in ℓ_1 using Cauchy random projections. *Journal of Machine Learning Research*, 8:2497–2532, 2007.

[20] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. Freeman. SIFT flow: dense correspondence across different scenes. In *ECCV*, 2008.

[21] D. Lowe. Object recognition from local scale-invariant features. In *ICCV*, 1999.

[22] S. Maji, A. Berg, and J. Malik. Classification using intersection kernel support vector machines is efficient. In *CVPR*, 2008.

[23] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *IJCV*, 65(1–2):43–72, 2005.

[24] S. Nene, S. Nayar, and H. Murase. Columbia object image library (COIL-100). Technical report, Columbia University CUCS-006-96, 1996.

[25] D. Nistér and H. Stewénius. Scalable recognition with a vocabulary tree. In *CVPR*, 2006.

[26] M. Plumbley. Recovery of sparse representations by polytope faces pursuit. In *Proceedings of International Conference on Independent Component Analysis and Blind Source Separation*, pages 206–213, 2006.

[27] A. Quattoni, M. Collins, and T. Darrell. Transfer learning for image classification with sparse prototype representations. In *CVPR*, 2008.

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

[29] A. Thomas, V. Ferrari, B. Leibe, T. Tuytelaars, B. Schiele, and L. Van Gool. Towards multi-view object class detection. In *CVPR*, 2006.

[30] R. Tibshirani. Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society B*, 58(1):267–288, 1996.

[31] J. Tropp. Algorithms for simultaneous sparse approximation. *Signal Process*, 86:572–602, 2006.

[32] S. Vempala. *The random projection method*. American Mathematical Society, 2004.

[33] A. Yang, S. Maji, K. Hong, P. Yan, and S. Sastry. Distributed compression and fusion of nonnegative sparse signals for multiple-view object recognition. In *International Conference on Information Fusion*, 2009.

[34] C. Yeo, P. Ahammad, and K. Ramchandran. Rate-efficient visual correspondences using random projections. In *ICIP*, 2008.