Distributed Compression and Fusion of Nonnegative Sparse Signals for Multiple-View Object Recognition *

Allen Y. Yang, Subhransu Maji, Kirak Hong, Posu Yan, and S. Shankar Sastry
Department of EECS, University of California, Berkeley, CA 94720
{yang,smaji,hokira,pyan,sastry}@ eecs.berkeley.edu

Abstract – *Visual surveillance in complex urban environments requires an intelligent system to automatically track and identify multiple objects of interest in a network of distributed cameras. The ability to perform robust object recognition is critical to compensate adverse conditions and improve performance, such as multi-object association, visual occlusion, and data fusion with hybrid sensor modalities. In this paper, we propose an efficient distributed data compression and fusion scheme to encode and transmit SIFT-based visual histograms in a multi-hop network to perform accurate 3-D object recognition. The method harnesses an emerging theory of (distributed) compressive sensing to encode high-dimensional, nonnegative sparse signals via random projection, which is unsupervised and independent to the sensor modality. A multi-hop protocol then transmits the compressed visual data to a base-station computer, which preserves a constant bandwidth regardless of the number of active camera nodes in the network. Finally, the multiple-view object features are simultaneously recovered via ℓ^1-minimization as an efficient decoder. The efficacy of the algorithm is validated using up to four Berkeley CITRIC camera motes deployed in a realistic indoor environment. The substantial computation power on the CITRIC mote also enables fast compression of SIFT-type visual features extracted from object images.*

Keywords: Sparse representation, distributed object recognition, fusion, compressive sensing.

1 Introduction

Consider a surveillance system consisting of a network of distributed, wireless cameras is instrumented to monitor certain events of interest. Often, such a system operates in complex urban environments, where a large class of objects may be present in the scene, e.g., pedestrians, vehicles, buildings, and highways. In traditional computer vision, the problem of *object recognition* has been extensively studied to detect and annotate objects from single camera views. The functionality enables the camera network to identify and track individual objects, and can improve the performance of the surveillance system in crowded urban environments. Successful methods have been demonstrated in the past, including pedestrian detection [12], general object detection [1, 22] (e.g., vehicles, toys, and animals), and scene annotation [4, 17] (e.g., buildings and highways). A large body of these works have been based on analysis of certain local image patches that are robust/invariant to image scaling and visual occlusion, which are the two common adverse conditions in image-based object recognition. The local image patches are typically extracted by a viewpoint-invariant interest point detector [15] combined with a SIFT (Scale-Invariant Feature Transform) descriptor [2, 13].

The application of object recognition in distributed camera sensor networks presents several unique challenges, which have been largely ignored in traditional single-camera systems:

1. How to efficiently compress and transmit object features from multiple camera sensors to a base station?
2. How to associate the features of a common object in 3-D across multiple camera views?
3. How to harness the multi-view information about the object to improve the accuracy of the recognition?

In computer vision, considerable efforts have been demonstrated to study the last two problems, while the cameras are considered to be reliably connected to a central computer [11, 22, 23]. In this paper, however, we are focused on addressing the first problem: *distributed compression and fusion of multiple-view visual features*. Traditional distributed source coding in camera networks only concerns direct compression of images and videos, while the correlation among fixed camera views is considered crucial to reduce the total coding length of multiple-view images [7, 24]. In object recognition, representation of the object appearance (i.e., the SIFT features) becomes a more compact description of the object. Therefore, the system should not just compress and streamline the captured images to a computer, which is extremely inefficient and impractical in a

*This work was supported in part by ARO MURI W911NF-06-1-0076.

band-limited wireless network (such as the low-power IEEE 802.15.4 protocol). The integration of mobile processors and memory units with camera sensors has provided several viable smart camera sensor platforms, where the SIFT-type features can be directly extracted on the camera. In particular, a fast SIFT implementation, called SURF (Speeded-Up Robust Features) [2], has been recently ported to several smart phone platforms. We utilize a custom-built open-source smart camera platform called CITRIC [5] to implement this function in this paper.

Furthermore, we study a general distributed compression framework, where wireless sensors may not directly communicate with the base station. As a result, the observations need to be relayed via one or many other sensors in the network, i.e., a *multi-hop* network. We also impose that, in order to compress the visual features, no information exchange is allowed between different camera views to learn about the mutual information between the cameras. Typically, learning such information requires extensive human intervention and well-conditioned training data sampled during real-world deployment. Any change of the relative camera positions will also render such information inaccurate or invalid. Finally, we consider a versatile fusion method for relaying the compressed image features such that the total bandwidth in a multi-hop communication remains constant, regardless of the number of active cameras in the network.

So far, we have stated the premise of the problem. The conditions of the problem may seem incongruous to some practitioners. For example, without learning the mutual information about the joint distribution in multiple views, there seems not possible to effectively fuse multiple sensor outputs. Also, the data bandwidth typically should increase in a multi-hop network when more sensors become active and demand transmission of their output to the base station.

A careful study will show in theory and practice that fast solutions to the above challenging problems exist, which will be demonstrated on the CITRIC system. The method harnesses an emerging theory of compressive sensing to encode high-dimensional, nonnegative sparse signals via random projection, which is unsupervised and independent to the sensor modality. The multiple-view object features are simultaneously recovered via ℓ^1-minimization on the base station. We also note that the general theory developed should also apply to other distributed fusion applications beyond object recognition, where the types of sparse signals may represent the temperature, precipitation, sound, and vicinity magnetic field, to name a few.

2 Sparse Representation in Object Recognition

Before we develop the general theory for compression and fusion of visual features in Section 3, we first review the literature of object recognition in traditional single camera views, the SIFT descriptor, and the role of sparse representation.

Arguably, vision-based object recognition has been motivated by the remarkable ability of humans to recognize objects in visual perception, either for their survival needs or for performing social functions. One influential theory in human vision explains the object recognition function on the basis of decomposition of objects into constituent parts (i.e., distinctive image patches) [1, 18, 19, 21]. 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 can be inferred that one or many 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 [17]. Local features are called *codewords*. Each codeword can be shared among multiple object classes. Hence, the codewords from all object categories are 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 representation of a single object figure is then *sparse*. One popular representation of the object features computes the instances of the appearance of all the codewords in an image. Since only a small number of features are exhibited on a specific object, their values in the representation are positive integers (e.g., a car can be seen to have two to four wheels depending on the viewpoint), and the majority of the values w.r.t. the other features in the vocabulary should be zero. Such representation is called a *histogram* [4,7,17]. As a result, the histogram becomes a compact representation of the object(s) that appear in the image. The key observation of such histograms is that they are *high-dimensional*, *sparse*, and *nonnegative*.

In computer vision, the requirement for extracting *robust* visual codewords that form a vocabulary is that their shapes must adapt to the pre-images on the objects in 3-D under different viewpoints. Robust image regions are called *interest points* or *affine-invariant features* [2, 15]. They can be selected at distinctive locations of an image, such as corners, blobs, and T-junctions. The appearance of these features is considered robust to change in viewpoint, scale, and pixel intensity. Figure 1 shows the extracted interest points from two related view points of a toy object and their correspondence. The images are sampled from a public Columbia COIL-100 object image database [16]. Further, the pixel values and the local gradients around a small neighborhood of each interest point are quantized into a feature vector, based on the rules of SIFT. For example, each interest point is quantized into a 64-D real vector in the SURF algorithm. After quantization, the similarity between two interest points between the image pair can be measured based on a distance metric between their corresponding feature vectors, e.g., the Euclidean distance. Figure 2 shows the histograms of the extracted SURF features in Figure 1. We observe that the histograms indicate that certain SURF features are shared between the two views.

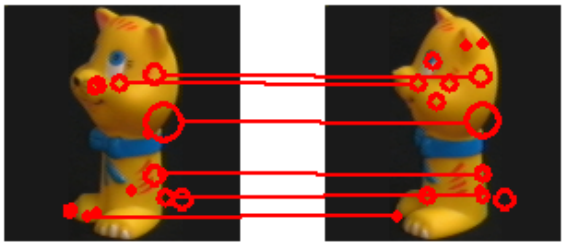

Figure 1: Detection of interest points (red circles) in two image views of a 3-D toy. The radius of each circle indicates the scale of the interest point in the image. The correspondence of the interest points that are invariant to viewpoint change is highlighted via red lines.

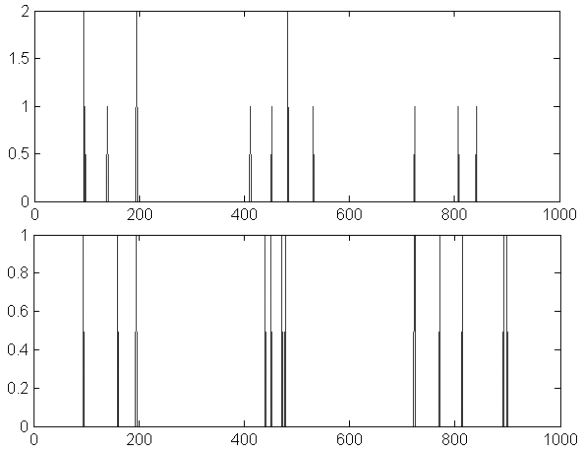

Figure 2: The histograms representing the image features from the two image views in Figure 1. The size of the vocabulary is 1000 based on the image features extracted from the 100 object classes in the entire COIL database. The two histograms are sparse and nonnegative.

3 Distributed Compression of Nonnegative Sparse Signals

In this section, we discuss the theory of distributed compression and fusion. We first define the problem:

Problem 1 (Distributed Compression and Fusion) *Suppose L sensor nodes independently observe a set of sparse signals $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_L \in \mathbb{R}^D$. The routing of the sensor network is organized as a tree structure: Each node may receive multiple output from other nodes as its children, but it can only send its output to a single node or the base station (root) as its parent.*

1. *On each node, construct an encoding function $f : \boldsymbol{x} \in \mathbb{R}^D \mapsto \boldsymbol{y} \in \mathbb{R}^d$ ($d < D$) that compresses the measurement signal.*

2. *On each parent node that receives l compressed features $\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_l$, including its own measurement, construct a fusion function $g : (\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_l) \mapsto \boldsymbol{y}' \in \mathbb{R}^d$. $\boldsymbol{y}'$ becomes the output of the parent node that encodes the information of its all child nodes. The fusion function maintains a constant dimension d in the output feature $\boldsymbol{y}'$, and hence the multi-hop bandwidth, regardless of the number of its child nodes.*

3. *On the base station, construct a decoding function that simultaneously recovers $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_L \in \mathbb{R}^D$.*

Several previous works have studied the problem of distributed data compression [9, 10, 23] and its application in object recognition [7]. In particular, [7] 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.

Inspired by the theory of compressive sensing, we propose a novel solution to compress and fuse multiple-view sparse histograms in a multi-hop sensor network. First, denote a D-dimensional histogram as $\boldsymbol{x}$. A linear transformation

$$f : \boldsymbol{y} = A\boldsymbol{x} \in \mathbb{R}^d, \tag{1}$$

reduces the dimension of $\boldsymbol{x}$ to d, where $A \in \mathbb{R}^{d \times D}$ is a projection matrix. Normally, A is always full-rank, and (1) represents an overcomplete linear system ($d < D$). Then the theory of CS [3] states that, for most full-rank matrices A, if $\boldsymbol{x}_0$ is sparse w.r.t. its dimension n, it is the unique solution of a regularized ℓ^0-minimization program

$$(P_0) \quad \min \|\boldsymbol{x}\|_0 \text{ subject to } \boldsymbol{y} = A\boldsymbol{x}. \tag{2}$$

Particularly, A can be a random projection matrix whose coefficients are drawn independently from a Gaussian distribution. To further simplify the implementation on low-power senor nodes, we also use a Bernoulli distribution of two values $(+1, -1)$ with even probability (i.e., the Rademacher distribution). Using random projection in the application of sensor networks warrants further comments. Compared with the other linear transformation functions, its main advantages are the following:

1. Random projection is efficient to generate directly on low-power sensors, and it does not depend on a domain-specific training set.

2. In terms of robustness to wireless congestion and packet loss, if (part of) the projected coefficients are lost 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 at a lower dimension in (1). 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.

3. In terms of security, if (part of) the projected coefficients are intercepted but the random seed used to generate the random matrix is not known to the intruder, it is more difficult to decipher the original signal than using other fixed filter banks.

Unfortunately, solving ℓ^0-minimization in general is an NP-hard problem, which requires an expensive combinatorial search over all possible subsets of nonzero coefficients. Hence, the bulk of the study in CS involves 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 ℓ^1-minimization:

$$(P_1) \quad \min \|\boldsymbol{x}\|_1 \text{ subject to } \boldsymbol{y} = A\boldsymbol{x}. \tag{3}$$

This relationship is called the ℓ^0/ℓ^1 equivalence.

In addition, it is important to note that $\boldsymbol{x}$ has to be nonnegative that represents the values of the object histogram. Hence, the (P_1) program is modified as

$$(P_1') \quad \min \|\boldsymbol{x}\|_1 \text{ subject to } \boldsymbol{y} = A\boldsymbol{x} \text{ and } \boldsymbol{x} \geq 0. \tag{4}$$

We study the effect of enforcing the nonnegativity to ℓ^0/ℓ^1 equivalence. First, for the triplet (k, d, D) that characterizes the relationship between the compression and the sparsity, we define $\delta = \frac{d}{D}$ and $\rho = \frac{k}{d}$, i.e., δ denotes the sampling rate and ρ denotes the relative sparsity w.r.t. the sampling dimension.

The equivalence relationship is closely connected to convex polytope theory, particularly in the high-dimension regime. Figure 3 illustrates a projection between a cross polytope $C \doteq C^3 \subset \mathbb{R}^3$ an 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-dimensional lower space. Clearly, some of the vertices and k-faces of C may be mapped to the interior of AC, i.e., they do not "survive" the projection.

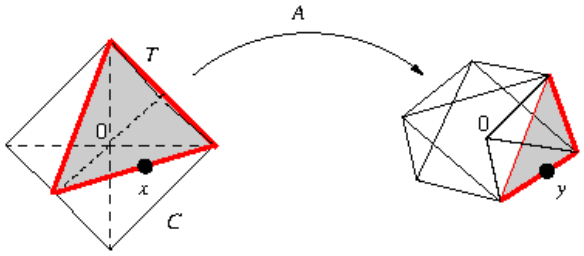

Figure 3: Projection of a cross polytope C in $\mathbb{R}^3$ to a quotient polytope AC via projection A. The corresponding simplex is T at the shaded area. Both AC and AT are 0-neighborly.

Compressive sensing provides the following equivalence condition for the ℓ^0/ℓ^1 equivalence relationship [8]:

Theorem 1 (ℓ^0/ℓ^1 Equivalence) *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 projection $A \in \mathbb{R}^{d\times D}$ and a $(k+1)$-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 1 is a powerful tool to examine if a sparse signal under a projection A can be uniquely recovered by ℓ^1-minimization. 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 ℓ^1-minimization.

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 result guarantees the well-behavior of random projection: In a high-dimensional space, with high probability, random projection A preserves most faces of a cross polytope. A short insight to this observation is that most randomly generated column vectors of A are linearly independent. To be more precise, with a fixed d and D, the upper bound for k is [8]

$$k \leq C\frac{d}{2e\log(D/(\sqrt{\pi}d))}, C \text{ is a constant.} \tag{5}$$

In object recognition, we consider a k-sparse vector $\boldsymbol{x}$ that is also nonnegative (assuming $\boldsymbol{x}$ is normalized to be ℓ^1-norm one). We denote $T \doteq T^{D-1}$ as the standard simplex in $\mathbb{R}^D$, i.e.,

$$T = \{\boldsymbol{x} : \|\boldsymbol{x}\| = 1 \text{ and } \boldsymbol{x} \geq 0\}. \tag{6}$$

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 naturally follows.

Theorem 2 (ℓ^0/ℓ^1 Equivalence of Nonnegative Signals) *Any nonnegative 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 all k-faces of T^{D-1} survive the projection A.*

The nonnegativity constraint reduces the domain of possible solutions for ℓ^1-minimization (as shown in Figure 3), and improves the ℓ^0/ℓ^1 equivalence relationship for a given (k, d, D). In particular, the upper bound for k for nonnegative signals is [8]

$$k \leq C\frac{d}{2e\log(D/(2\sqrt{\pi}d))}, C \text{ is a constant.} \tag{7}$$

Now we are ready to discuss the distributed fusion function. For a parent node that receives l inputs $(\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_l)$ from its child nodes, the fusion function is defined as

$$g: \quad \boldsymbol{y}' \doteq \boldsymbol{y}_1 + \boldsymbol{y}_2 + \cdots + \boldsymbol{y}_l \in \mathbb{R}^d, \\ = [A_1, A_2, \cdots, A_l] \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_l \end{bmatrix}, \tag{8}$$

where $A_1, A_2, \cdots, A_l$ are the respective random projection matrices on the l sensor nodes, and $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_l$ are the corresponding histogram vectors. In theory, the fusion rule (8) does not lose information about the l histograms, given that the ℓ^0/ℓ^1 equivalence still holds to solve the following program

$$\min \|\boldsymbol{x}'\|_1 \text{ subject to } \boldsymbol{y}' = A'\boldsymbol{x}' \text{ and } \boldsymbol{x}' \geq 0, \tag{9}$$

where $A' = [A_1, A_2, \cdots, A_l] \in \mathbb{R}^{d \times lD}$ and $\boldsymbol{x}' = \left[\boldsymbol{x}_1^T, \boldsymbol{x}_2^T, \cdots, \boldsymbol{x}_l^T\right]^T \in \mathbb{R}^{lD}$.

In practice, (8) provides an adaptive solution to fuse multiple sensor data given by random projection: Regardless of the number of active child nodes, the implementation of the fusion rule is straightforward. In general, each random matrix A_i should be generated using a distinctive random seed, such that there is no ambiguity in recovering the histogram signal $\left[\boldsymbol{x}_1^T, \boldsymbol{x}_2^T, \cdots, \boldsymbol{x}_l^T\right]^T$. On the station side, after the computer receives $l = L$ compressed feature(s) from its child nodes, all the sparse histograms can be simultaneously recovered via (9).

In Section 4, the performance of the algorithm and the system implementation will be validated. In particular, the experiment using the real data sampled from the COIL database shows that the accuracy in (9) grows proportionally to the decrease in the total number L of the cameras and to the increase in the dimension d of the projection. For example, in Figure 6, the ℓ^1 reconstruction error for a single camera at $d = 100$ is equal to that for two cameras at $d = 200$, and is also equal to that for three cameras at $d = 300$ and so forth. Hence, given the fusion condition to preserve the same bandwidth, the rule (8) performs *equally well* as proportionally changing each histogram projection d based on the number of active sensors.

Finally, we outline a nonnegative orthogonal matching pursuit (NOMP) algorithm to solve the problem (P_1') (Algorithm 1). It is modified from the orthogonal matching pursuit (OMP) algorithm for general sparse signals. Many other sparse solvers can be regarded as extensions of OMP (e.g., basis pursuit (BP) [6] and polytope faces pursuit [20]), and they can be also modified to take into account of the nonnegativity condition. For example, A nonnegative interior-point method is discussed in [14]. All the ℓ^1-minimization operated on the base station in this paper is implemented in MATLAB using the *cvx* package.[1]

[1] `http://www.stanford.edu/~boyd/cvx/`.

Algorithm 1 Nonnegative Orthogonal Matching Pursuit

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

1: Initialization: $k \leftarrow 0$. Assign residual $\boldsymbol{r}_0 \leftarrow \boldsymbol{y}$, a sparse support index set $\Omega \leftarrow \emptyset$ (define Ω^c as its complement), and $\boldsymbol{x}_0 \leftarrow 0 \in \mathbb{R}^D$.
2: **repeat**
3: $k \leftarrow k + 1$.
4: $i = \arg\max_{j \in \Omega^c} \frac{\boldsymbol{v}_j^T \boldsymbol{r}_{k-1}}{\|\boldsymbol{v}_j\|_2}$.
5: **if** $\boldsymbol{v}_i^T \boldsymbol{r}_{k-1} > 0$ **then**
6: Add i to the support set: $\Omega = \Omega \cup \{i\}$.
7: **else**
8: No other possible vertex to add. **break**.
9: **end if**
10: Estimate the coefficients in support Ω:

$$\boldsymbol{x}_k^\Omega \leftarrow (A^\Omega)^\dagger \boldsymbol{y},$$

where $A^\Omega = [\cdots \boldsymbol{v}_i \cdots]$, $i \in \Omega$.
11: $\boldsymbol{r}_k \leftarrow \boldsymbol{y} - A\boldsymbol{x}_k$.
12: **until** $\|\boldsymbol{r}_k\|_2 < \epsilon$.

Output: $\boldsymbol{x}^* \leftarrow \boldsymbol{x}_k$.

4 Experiment

We first discuss the implementation of the multihop routing protocol using up to four CITRIC camera motes [5] and a Linux laptop as the base station computer. 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.

The multihop routing is based on the *collection tree protocol* (CTP), which is a tree-based address-free multihop routing protocol. The CTP is chosen for our experiment due to its high data throughput compared to other multihop protocols currently available in TinyOS. For our experiment, we have modified the routing engine of the CTP to fix the routing paths from the CITRIC motes to the base station such that we can study the behavior of the network at fixed numbers of hops. With four CITRIC motes, we design a chain hierarchy with a single leaf note and a single root note (i.e., the base station). Except for the leaf node, each of the other three camera motes has one child and one parent. Note that the distributed compression algorithm also works on other types of tree-based multihop configurations. Given a fixed number of sensors, the chain structure necessarily produces maximal network latency.

When an EXECUTE command is issued from the base station, the camera motes relay the command based on the order of the routing protocol. Upon receiving the command, each sensor capture an image from the camera sensor, produce a nonnegative histogram vector, and compress the vector to a low-dimensional feature vector. In this experiment,

we choose a 1000-codeword vocabulary constructed from the COIL database. The vocabulary is preloaded on the sensor nodes before the experiment. The tested random projection dimensions range from 100 to 500. Once the leaf mote has calculated the histogram vector, it becomes the first node to send the vector to its parent. The parent will add the received vector(s) with its own vector, and then output to its parent.

The first experiment measures the average latency of the whole process, as shown in Figure 4. It is conducted at multiple locations of an office building to cover six typical indoor scenes/objects: an office desk, a bookshelf, an indoor court yard, a student lounge, a tree, and a corridor. Two examples of the images are shown in figure 5. Under each network configuration, the experiment repeats 10 times. Except for the distributed fusion step (8) that depends on the multihop protocol, the remaining functions to produce the visual features are operated in parallel. First, the algorithm takes about 15 sec on a single-node CITRIC network, which is the baseline performance without the multihop component. In the presence of multiple cameras in the network, the latency is about 20 sec. The difference of the latency among different active cameras is small, which indicates that the bottleneck of the algorithm is the SURF feature detection.

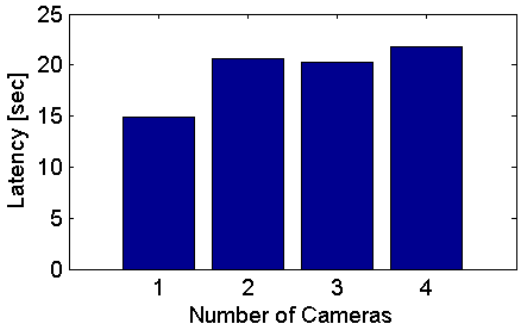

Figure 4: The average latency of distributed object feature extraction with one to four CITRIC motes.

Figure 5: A corridor scene (left) and a tree object (right). The images are grayscale at 320×240 resolution. The SURF features are superimposed as red circles.

The second experiment measures the accuracy of the encoding/decoding algorithm. The comparison is based on two performance indices: 1. The traditional ℓ^2-norm distortion per histogram. 2. The sparse support error per histogram, i.e., the difference in the locations of the nonzero coefficients between the original histogram and the recovered one. As an abstraction of an image, the histogram reveals the types of the visual features in the scene via its sparse support, and the repetition of the visual features via the values in the histogram. Both the sparse support and its values are equally important in recognizing the object(s) in the image.

In this experiment, we choose to use the multiple-view images from the publicly available COIL database, such that the results are reproducible. All the images in COIL are RGB color images with 128×128 resolution. In computing their SURF features, they are converted to grayscale images of the same resolution. Images of the same object from COIL are randomly sampled and loaded onto the four CITRIC motes to simulate the imaging process. Given a precomputed 1000-codeword vocabulary, the true histograms w.r.t. a fixed SURF implementation can be obtained, and they are used as the ground truth to compare with the estimated histograms from the base station via (9).

Figure 6 first show the performance using NOMP. With a single camera, NOMP perfectly recovers 1000-D histograms via 200-D projection (i.e, $\delta = 0.2$). With two cameras, the length of the joint histogram clearly doubles. The curves show that 400-D projection perfectly recovers the joint histogram, which is about twice the size of the previous projection. The difference in performance is consistent with the other camera configurations. The maximal sparse support error is about three, meaning that over 1000 coefficients in a histogram, the average number of difference locations of nonzero coefficients is three, which is extremely small. When the projection is increased to 300-D, the maximal sparse support error is reduced to below two.

Finally, we validate that NOMP outperforms the OMP algorithm (3) when the underlying sparse signal is indeed nonnegative, which is shown in Figure 7. OMP achieves perfect recovery from a single camera at 300-D projection, and from two cameras at 500-D projection, which are both 100-D higher than NOMP. Overall, NOMP improves the performance w.r.t. both norm distortion and sparse support by about one third under the same setup.

5 Conclusion

We have addressed a distributed compression and fusion problem in transmitting nonnegative sparse signals to represent visual objects from multiple camera views. The key observation is that, under a large visual vocabulary, an object histogram obtained by tallying SIFT-type codewords should be nonnegative and sparse. We have considered the scenario where images of a common 3-D object are captured from multiple camera sensor nodes, and the goal is to effectively compress and transmit the visual histograms via a low-bandwidth multihop network. We have demon-

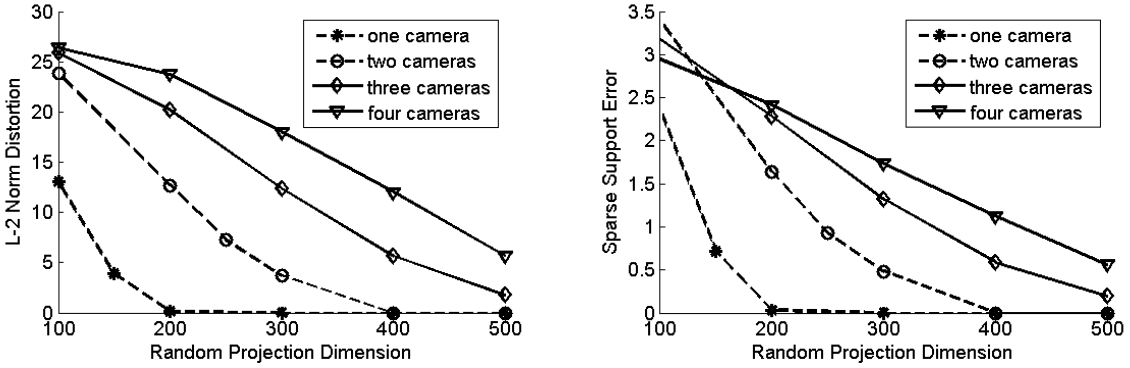

Figure 6: Distortion of the distributed coding scheme via NOMP. Left: Norm distortion. Right: Sparse support error.

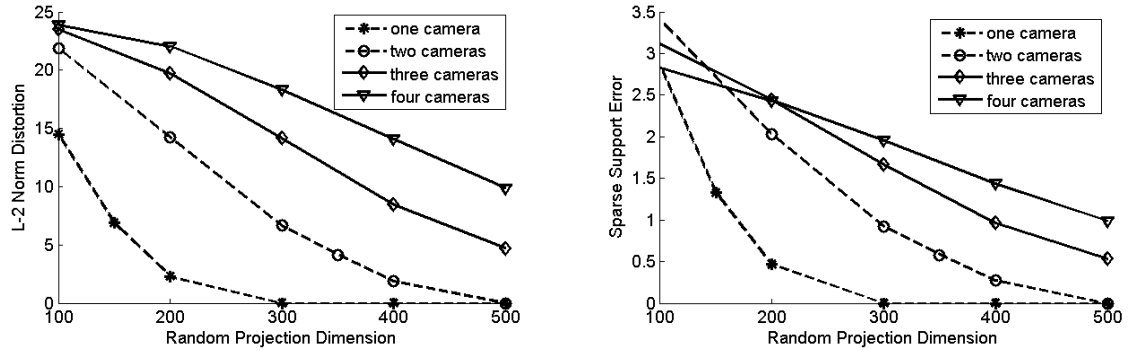

Figure 7: Distortion of the distributed coding scheme via OMP. Left: Norm distortion. Right: Sparse support error.

strated a highly flexible compression and fusion scheme based on random projection. Under the constant bandwidth constraint, we have shown that random projection performs *equally well* as proportionally changing each histogram projection based on the number of active sensors. At the base station, the multiple-view histograms are simultaneously recovered by nonnegative ℓ^1-minimization routines. The efficacy of the method has been validated via both simulation and a real-world experiment using the CITRIC smart camera motes.

The paper has also raised several important open problems in distributed object recognition. In particular, given the resource-constrained nature of the camera sensors, how to detect and locate multiple objects in a 3-D scene; how to establish feature correspondences among different camera views; and how to harness the rich information of multiple-view features to improve the recognition accuracy at the base station? Future advanced sensor surveillance systems for complex urban environments demand a more comprehensive distributed framework to address these questions.

Acknowledgments

The authors would like to thank Dr. Trevor Darrell and Mario Christoudias of the University of California, Berkeley, for their valuable suggestions.

References

[1] S. Agarwal and D. Roth. Learning a sparse representation for object detection. In *Proceedings of the European Conference on Computer Vision*, 2002.

[2] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool. SURF: Speeded up robust features. *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[3] A. Bruckstein, D. Donoho, , and M. Elad. From sparse solutions of systems of equations to sparse modeling of signals and images. *(in press) SIAM Review*, 2007.

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

[5] 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.

[6] S. Chen, D. Donoho, and M. Saunders. Atomic decomposition by basis pursuit. *SIAM Review*, 43(1):129–159, 2001.

[7] C. Christoudias, R. Urtasun, and T. Darrell. Unsupervised feature selection via distributed coding for multi-view object recognition. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2008.

[8] D. Donoho and J. Tanner. Counting faces of randomly-projected polytopes when the projection radically lowers dimension. *Journal of the American Mathematical Society*, 22(1):1–53, 2009.

[9] 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.

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

[11] V. Ferrari, T. Tuytelaars, and L. V. Gool. Integrating multiple model views for object recognition. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2004.

[12] B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pages 878– 885, 2005.

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

[14] F. Malgouyres and T. Zeng. A predual proximal point algorithm solving a nonnegative basis pursuit denoising model. *print*, 2007.

[15] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool. A comparison of affine region detectors. *International Journal on Computer Vision*, 65(1–2):43–72, 2005.

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

[17] 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.

[18] B. Olshausen and D. Field. Natural image statistics and efficient coding. *Network: Computation in Neural Systems*, 7:333–339, 1996.

[19] M. Oram and D. Perrett. Modeling visual recognition from neurobiological constraints. *Neural Networks*, 7:945–972, 1994.

[20] 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.

[21] M. Riesenhuber and T. Poggio. CBF: A new framework for object categorization in cortex. In *Proceedings of International Workshop on Biologically Motivated Computer Vision*, pages 1–9, 2000.

[22] A. Thomas, V. Ferrari, B. Leibe, T. Tuytelaars, B. Schiele, and L. V. Gool. Towards multi-view object class detection. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2006.

[23] 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.

[24] C. Yeo and K. Ramchandran. Robust distributed multiview video compression for wireless camera networks. In *Proceedings in Visual Communications and Image Processing*, 2007.