Towards an efficient distributed object recognition system in wireless smart camera networks *

Nikhil Naikal, Allen Y. Yang, and S. Shankar Sastry
Department of EECS, University of California, Berkeley, CA 94720
{nnaikal,yang,sastry}@ eecs.berkeley.edu

Abstract – *We propose an efficient distributed object recognition system for sensing, compression, and recognition of 3-D objects and landmarks using a network of wireless smart cameras. The foundation is based on a recent work that shows the representation of scale-invariant image features exhibit certain degree of sparsity: If a common object is observed by multiple cameras from different vantage points, the corresponding features can be efficiently compressed in a distributed fashion, and the joint signals can be simultaneously decoded based on distributed compressive sensing theory. In this paper, we first present a public multiple-view object recognition database, called the Berkeley Multiview Wireless (BMW) database. It captures the 3-D appearance of 20 landmark buildings sampled by five low-power, low-resolution camera sensors from multiple vantage points. Then we review and benchmark state-of-the-art methods to extract image features and compress their sparse representations. Finally, we propose a fast multiple-view recognition method to jointly classify the object observed by the cameras. To this end, a distributed object recognition system is implemented on the Berkeley CITRIC smart camera platform. The system is capable of adapting to different network configurations and the wireless bandwidth. The multiple-view classification improves the performance of object recognition upon the traditional per-view classification algorithms.*

Keywords: Distributed object recognition, compressive sensing, smart camera networks.

1 Introduction

Distributed object recognition is a fast-growing research topic [7, 8, 11, 20, 24, 26], mainly motivated by the proliferation of portable camera devices and their integration with modern wireless sensor network technologies. Given a wireless network of cameras, the new paradigm studies how to classify a 3-D object that may be captured from multiple vantage points. The ability to acquire multiple-view observations of a common object can effectively compensate many visual nuisances such as object occlusion and pose variation, and may further boost the recognition accuracy if the multiple-view images are properly utilized.

Recent studies in distributed object recognition can be summarized in three intimately related areas. The first area is focused on the development of smart camera platforms. In recent years, several experimental platforms have successfully integrated high-resolution cameras (together with other sensing modalities) with state-of-the-art mobile processors and considerable amounts of memory. The reader is referred to [6] for more details in this area.

The second area concerns the extraction of dominant image features to represent the 3-D objects that are captured in the images. Leveraging on the available processing power of many smart cameras, these image features can be directly extracted on the camera sensor without relaying the full-resolution images to a base-station computer. Then, the choice of optimal object features for particular applications boils down to two factors: on one hand, the efficiency to compute these image features on the smart sensor; on the other hand, the accuracy to concisely represent the 2-D appearance of the objects. The success of SIFT-type viewpoint invariant feature detectors [13, 14] has led to the development of other improved feature detectors and descriptors, such as SURF [2] and CHoG [5], which are better suited for deployment on mobile camera platforms.

The third area concerns the correspondence and compression of image features extracted from the multiple camera views. In a per-view basis, [26] argued that reliable feature correspondence can be established in a much lower-dimensional space between camera sensors, even if the feature vectors are linearly projected onto a random subspace. With multiple camera views, [8] studied a SIFT-feature selection algorithm, where the number of SIFT features that need to be transmitted to the base station can be reduced

*This work was supported in part by ARO MURI W911NF-06-1-0076 and ARL MAST-CTA W911NF-08-2-0004. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute for Government purposes notwithstanding any copyright notation hereon.

by considering the joint distribution of the features among multiple camera views of a common object. A recent work [21] further considered using robust structure-from-motion techniques (e.g., RANSAC) to select strong object features between two camera views that satisfy an epipolar constraint induced by a large baseline transformation, and subsequently reject weak features as outliers from the final stage of object recognition.

Contributions. We present a systematic study on distributed object recognition in low-power wireless smart camera networks. The work is based on an open-source smart camera platform, called CITRIC [6], which integrates a high-resolution camera with a 600 MHz fixed-point mobile processor and 80 MB memory. First, we propose a new multiple-view object database as a public platform to benchmark the system, which is referred to as the Berkeley Multiview Wireless (BMW) database. We assume the camera sensors and the network station are connected only through a band-limited wireless channel. Motivated by the emerging theory of compressive sensing (CS), we overview a sparsity-based distributed sampling scheme to compress certain feature histograms that concisely represent the appearance of a common object in multiple views [24]. The discussion also covers the most recent development in CS to effectively recover sparse signals using fast ℓ_1-minimization (ℓ_1-min) algorithms. Finally, we propose a multiple-view recognition method to jointly classify the object observed by multiple cameras in the network, a concept that has been largely ignored by existing solutions. We show that the multiple-view classification significantly improves the performance upon traditional per-view classification algorithms in both small-baseline and large-baseline situations. Furthermore, the system is capable of adapting to the change in different network configurations and the wireless bandwidth.

2 Berkeley Multiview Wireless Database

In the literature, there exist several public image-based object recognition databases, such as Oxford Buildings [17], COIL-100 [15], and Caltech-101 [10]. However, most of the databases are constructed using high-resolution cameras that do not take into account the real-world noise and distortion exhibited by most low-power camera sensors in surveillance applications. In addition, some databases only capture object images in lab-controlled indoor environments (such as COIL-100), while others collect a wide variety of object images in the same categories that may not necessarily share the same appearance in 3-D (such as Caltech-101). To aid peer evaluation of distributed object recognition methods for the wireless surveillance scenario, we have constructed a public multiple-view image database, namely, the BMW database. The database can be accessed online at: `http://www.eecs.berkeley.edu/~yang/software/CITRIC/`.

The BMW database consists of multiple-view images of 20 landmark buildings on the campus of University of California, Berkeley. For each building, 16 different vantage points have been selected to measure the 3-D appearance of the building. The apparatus for image acquisition incorporates five low-power CITRIC camera sensors [6] on a tripod, which can be triggered simultaneously. Figure 1 shows the configuration of the camera apparatus. Figure 2 shows some examples of the captured building images. The cameras on the periphery of the cross are named Cam 0, Cam 1, Cam 4, Cam 3 with a counter-clockwise naming convention, and the center camera is named Cam 2. Thus, the BMW database has a total of 960 images.

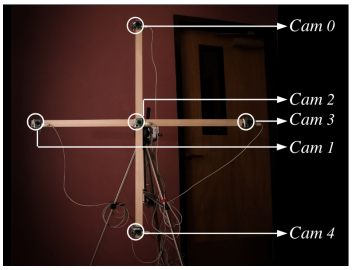

Figure 1: The apparatus that instruments five camera sensors.

(a) Five small-baseline images captured at one vantage point.

(b) Five large-baseline images captured at different vantage points.

Figure 2: Examples of multiple-view images of a building (the Campanile at UC Berkeley) in the BMW database.

It is worth emphasizing the following properties of the BMW database:

1. The images have been captured outdoor in different sessions. Therefore, some variations in ambient illumination exist within each building category and across different categories.

2. The image quality is considerably lower than many existing high-resolution databases, which is intended to reproduce realistic imaging conditions for camera surveillance applications. All images are 640×480 RGB color images. Since the CITRIC camera sensor does not have an auto-focus mechanism, the focal length of the camera is permanently set to maximum. However, it is noticeable that some images are slightly out of focus. In some cases, small image regions are visibly corrupted by dust residual on the camera lenses.

3. More importantly, the database provides a two-tier multiple-view relationship to systematically benchmark the performance of multiple-view object recognition algorithms, as shown in Figure 2. Specifically,

the five images sampled at each vantage point simulate small-baseline camera transformations, while the images sampled at different vantage points simulate large-baseline camera transformations. Furthermore, the small-baseline image sets can be used to simulate the scenario where a slowly moving camera continuously sample images in a short time frame. In Section 5, we will systematically examine the recognition performance in both small-baseline and large-baseline scenarios.

3 Encoding Multiple-View Object Images via A Joint Sparsity Model

In this section, we briefly review a sparsity-based sampling scheme [24] to encode useful information in multiple-view object images from a distributed camera network. To implement a fast codec to recover distributed source signals in a sensor network setup, we also discuss the latest results on accelerated ℓ_1-min algorithms in the CS and optimization literature [23].

3.1 The Joint Sparsity Model

We assume multiple cameras are equipped to observe a common 3-D scene from different vantage points. For distributed object recognition, it is reasonable to simplify the communication model between sensors and the base station as a single-hop wireless network, i.e., the topology of the network is a star shape with the computer at the center and all the sensors directly communicate to the computer.

Using a SIFT-type feature detector, certain viewpoint-invariant features can be extracted from the images, as shown in Figure 3. For an object database (e.g., BMW), object features may be shared between different object classes. Therefore, all features extracted from the training images can be clustered/quantized based on their visual similarities into a vocabulary. The clustering normally is based on a hierarchical k-means algorithm [12]. The size of a vocabulary for a large database ranges from thousands to hundreds of thousands. For example, in this paper, we use hierarchical k-means to construct 10,000-D vocabularies for the BMW database, with $k = 10$ and four hierarchies. Figure 4 shows the 10,000-D vocabulary tree constructed using all the CHoG features from the BMW training set (see Section 5 for more detail).

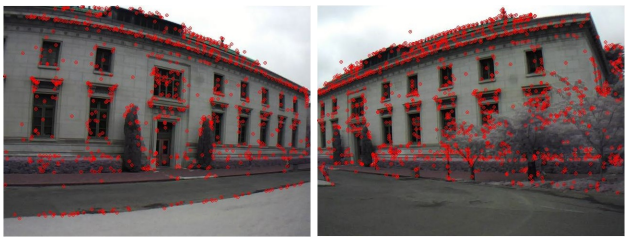

Figure 3: CHoG feature points detected in a pair of image views of a building.

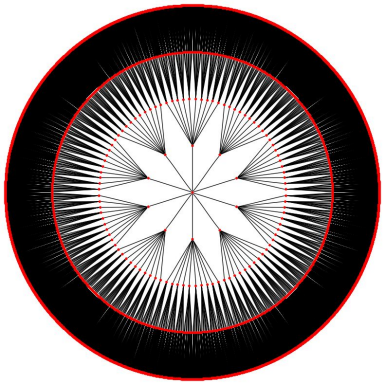

Figure 4: The 10,000-D vocabulary tree built using all CHoG features extracted from the training images in the BMW database. The tree is radially represented, with the center being the root node.

In [24], the authors have argued that, given a large vocabulary that contains quantized SIFT features from many classes, the representation of the features extracted from a single image is *sparse*, which is called a SIFT histogram. If we denote L as the number of the camera sensors that observe the same object in 3-D, and $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_L \in \mathbb{R}^D$ are the corresponding SIFT histogram vectors. Then each coefficient in $\boldsymbol{x}_i$ represents the instances of one type of the SIFT feature in the i-th view. Since only a small number of the features may be exhibited in a single image, the majority of the histogram coefficients should be (close to) zero. More importantly, since SIFT-type features are robust to some degree of camera rotation and translation, images from different vantage points may share a subset of the same features, thus yielding histograms with similar coefficient values, as shown in Figure 5.

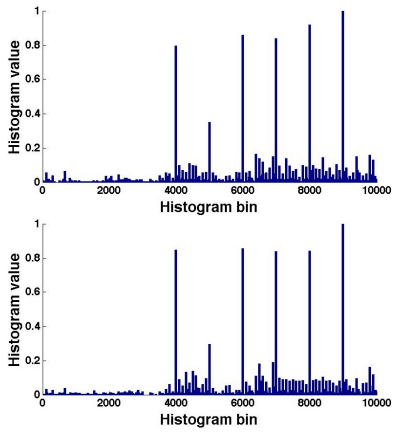

Figure 5: The 10,000-D feature histograms corresponding to the image pair in Figure 3. The joint sparsity patten indicates certain dominant features are shared between the two views.

The problem of encoding multiple-view object images can be formulated as the following. For the high-dimensional histogram vectors extracted from the L images,

define a *joint sparsity* (JS) model as

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

where $\boldsymbol{x}_c$ represents the *common* component, and each $\boldsymbol{z}_i$ represents an *innovation*. Furthermore, both $\boldsymbol{x}_c$ and $\boldsymbol{z}_i$ are also sparse. On each camera sensor, an encoding function $\boldsymbol{b}_i \doteq f(\boldsymbol{x}_i) \in \mathbb{R}^d$ is sought to compress the histogram vector $\boldsymbol{x}_i$. At the base station, upon receiving $\boldsymbol{b}_1, \cdots, \boldsymbol{b}_L$ compressed features, the system should simultaneously recover the source signals $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_L$, and further proceed to classify the 3-D object represented by the multiple-view histograms.

3.2 Distributed Encoding of JS Signals

The fact that each $\boldsymbol{x}_i$ is sparse against a large vocabulary provides a means to effectively sample the signal via a linear projection, motivated by the CS theory. In particular, we define a *random* matrix $A \in \mathbb{R}^{d \times D}$ as an overcomplete dictionary (i.e., $d < D$) whose elements are sampled from independent and identically-distributed Gaussians. Then a random projection function is defined as:

$$f : \boldsymbol{b} = A\boldsymbol{x}. \tag{2}$$

However, recovering $\boldsymbol{x}$ from (2) essentially is an inverse problem, as the number of observations in $\boldsymbol{b}$ is smaller than the number of unknowns in $\boldsymbol{x}$. The CS theory [4, 9] shows that if the underlying signal $\boldsymbol{x}_i$ is sufficiently sparse and the projection dimension $d > \delta(A)D$ is above a threshold determined by $\delta(A)$, then $\boldsymbol{x}_i$ is the unique solution to a convex program called ℓ_1-min:

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

In other words, (P_1) guarantees that no information is lost by projecting $\boldsymbol{x}_i$ onto a low-dimensional random subspace, as long as $\boldsymbol{x}_i$ is sufficiently sparse.

Now we can consider the decoding problem at the base station. Given the fact that all camera views may share a sparse component $\boldsymbol{x}_c$, the ensemble $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_L$ can be simultaneously recovered at the base station with the accuracy that may exceed that by estimating (P_1) individually [24]. In particular, the JS model can be solved in a single linear system:

$$\begin{aligned} \begin{bmatrix} \boldsymbol{b}_1 \\ \vdots \\ \boldsymbol{b}_L \end{bmatrix} &= \begin{bmatrix} A_1 & A_1 & 0 & \cdots & 0 \\ \vdots & & \ddots & \ddots & \\ A_L & 0 & \cdots & 0 & A_L \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_c \\ \boldsymbol{z}_1 \\ \vdots \\ \boldsymbol{z}_L \end{bmatrix} \\ \Leftrightarrow \qquad \boldsymbol{b}' &= A'\boldsymbol{x}' \in \mathbb{R}^{dL}. \end{aligned} \tag{4}$$

Enforcing the JS model can boost the estimation accuracy in (P_1) when $d_1 = d_2 = \cdots = d_L = d$ is uniform. More importantly, it also makes it possible to choose different sampling rates for individual camera sensors. This property is particularly relevant to wireless sensor networks, where sensor nodes that have lower bandwidth or lower power reserve may choose to reduce their sampling rates in order to preserve energy.

More specifically, the strategy of choosing varying sampling rates can be viewed as an application of the celebrated Slepian-Wolf theorem [19]. For the simplest case of two source channels X_1 and X_2, the theorem shows that, given sequences x_1 and x_2 that are generated from the two channels respectively, the sequences can be jointly recovered with vanishing error probability asymptotically *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.

In distributed object recognition, with the JS model, a *necessary* condition for simultaneously recovering $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_L$ can be found in [1]. Basically, it requires that each sampling rate $\delta_i = \frac{d_i}{D}$ guarantees the so-called *minimal sparsity signal* of $\boldsymbol{z}_i$ is sufficiently encoded, and also the total sampling rate guarantees that both the joint sparsity and the innovations are sufficiently encoded.

3.3 Decoding Sparse Signals via Fast ℓ_1-Minimization Algorithms

Finally, we briefly discuss the state of the art in effectively solving the convex program (P_1) via an accelerated ℓ_1-min technique. A comprehensive review of existing fast ℓ_1-min algorithms can be found in [23].

The convex program (P_1) has traditionally been formulated as a linear programming problem called *basis pursuit* (BP), which has several well-known solutions via iterative interior-point methods. However, the computational complexity of these interior-point methods is often too high for many real-world, large-scale applications. The main reason is that they all involve expensive operations such as matrix factorization and solving linear least squares.

Recently, *iteraive shrinkage-thresholding* (IST) methods have been recognized as a good alternative to the exact BP solutions. The approach is also appealing to large-scale applications because its implementation mainly involves lightweight operations such as vector operations and matrix-vector multiplications, which is in contrast to most past ℓ_1-min algorithms.

In a nutshell, IST considers a variation of (P_1) that takes into account the existence of measurement errors in the sensing process:

$$(P_{1,2}): \quad \min \|\boldsymbol{x}\|_1 \text{ subject to } \|\boldsymbol{b} - A\boldsymbol{x}\|_2 \leq \epsilon, \tag{5}$$

where ϵ is a bound on the additive white noise in $\boldsymbol{b}$. By the Lagrangian method, $(P_{1,2})$ is rewritten as an unconstrained *composite objective function*:

$$\min_{\boldsymbol{x}} F(\boldsymbol{x}) \doteq \frac{1}{2}\|\boldsymbol{b} - A\boldsymbol{x}\|_2^2 + \lambda\|\boldsymbol{x}\|_1, \tag{6}$$

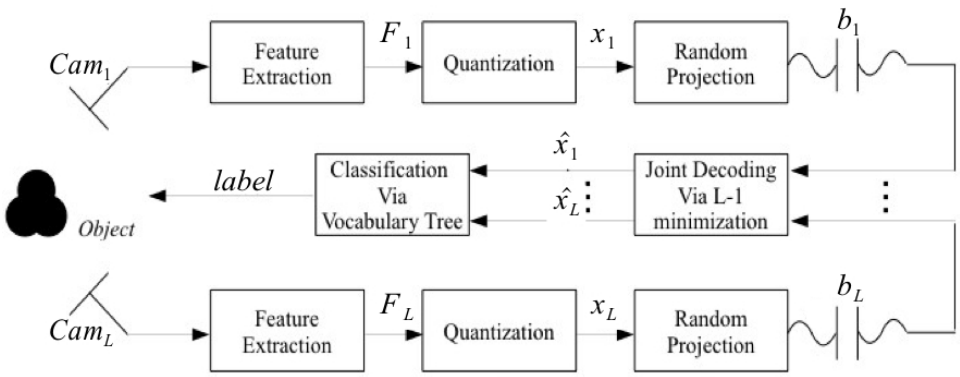

Figure 6: Flow diagram of the sparsity-based distributed object recognition system.

where $\lambda > 0$ is the Lagrangian multiplier.

We can immediately see that the main issue in optimizing such a composite function $F(\boldsymbol{x})$ is that its second term $\|\boldsymbol{x}\|_1$ is not a smooth function and therefore is not differentiable everywhere. Nevertheless, one can always locally linearize the objective function in an iterative fashion as [3, 22]:

$$\begin{aligned} \boldsymbol{x}^{(k+1)} &= \arg\min_{\boldsymbol{x}}\{f(\boldsymbol{x}^{(k)}) + (\boldsymbol{x}-\boldsymbol{x}^{(k)})^T\nabla f(\boldsymbol{x}^{(k)}) \\ &\quad +\tfrac{1}{2}\|\boldsymbol{x}-\boldsymbol{x}^{(k)}\|_2^2 \cdot \nabla^2 f(\boldsymbol{x}^{(k)}) + \lambda g(\boldsymbol{x})\} \\ &\approx \arg\min_{\boldsymbol{x}}\{(\boldsymbol{x}-\boldsymbol{x}^{(k)})^T\nabla f(\boldsymbol{x}^{(k)}) \\ &\quad +\tfrac{\alpha^{(k)}}{2}\|\boldsymbol{x}-\boldsymbol{x}^{(k)}\|_2^2 + \lambda g(\boldsymbol{x})\}, \end{aligned} \tag{7}$$

where the hessian $\nabla^2 f(\boldsymbol{x}^{(k)})$ is approximated by a diagonal matrix $\alpha^{(k)} I$.

One can further show that the linearized objective function (7) has a closed-form solution called the *soft-thresholding* function [3, 22]. Furthermore, the speed of convergence from an initial guess $\boldsymbol{x}^{(0)}$ to the ground-truth sparse signal can be *accelerated* by a numerical technique called the *alternating direction method* (ADM) [25]. For ℓ_1-min, ADM iteratively optimizes both the sparse signal $\boldsymbol{x}$ and the residual term $\boldsymbol{e}$:

$$\min_{\boldsymbol{x},\boldsymbol{e}}\{\|\boldsymbol{x}\|_1 + \frac{1}{2\mu}\|\boldsymbol{e}\|^2 + \frac{1}{2\lambda}\|A\boldsymbol{x}+\boldsymbol{e}-\boldsymbol{b}\|^2 - \boldsymbol{y}^T(A\boldsymbol{x}+\boldsymbol{e}-\boldsymbol{b})\}, \tag{8}$$

where $\boldsymbol{y} \in \mathbb{R}^d$ and $\mu > 0$ are two additional variables. It is easy to see that when $\boldsymbol{e}$ is fixed, (8) can be converted to the standard IST problem in (7); when $\boldsymbol{x}$ is fixed, since the ℓ_1-norm $\|\boldsymbol{x}\|_1$ becomes a constant, the objective function becomes smooth and its optimum is trivial to compute.

4 Multiple-View Object Recognition using a Hierarchical Vocabulary Tree

In this section, we explain an efficient multiple-view object recognition algorithm that takes multiple-view histograms as the input, and outputs a label as the classification of the object in 3-D. Figure 6 summarizes the complete system diagram.

Given a large set of robust image features (e.g., SIFT), we can construct a vocabulary tree using hierarchical *k*-means, where *k* represents the branch factor of the tree [16]. On the highest level of the tree, all the feature descriptors are partitioned into *k* clusters, with the mean of each cluster representing the cluster center. At each lower level, *k*-means is applied within each previous cluster in order to further partition the space into *k* clusters. The process is continued until there are k^H clusters at the *H*-th level (as shown in Figure 4).

With the vocabulary tree constructed, the feature descriptors in each training image are propagated down the tree. Then a term-frequency inverse-document-frequency (*tf-idf*) weighted histogram $\boldsymbol{y}$ can be defined for each training image as follows. First, assign an entropy-based weight w_p to each quantized leaf node feature p in the vocabulary tree as

$$w_p \doteq \log\frac{N}{N_p}, \tag{9}$$

where N is the total number of the training images, and N_p is the number of training images that contain the same feature vector p. With the weight w_p computed in this manner, all the elements of the training histograms $\boldsymbol{y}$ and test histograms $\boldsymbol{x}$ are multiplied element-wise with this weight function in order to achieve the *tf-idf* weighting scheme. For each object category, $i = 1 \cdots C$, multiple weighted histograms are generated for all m training images of that object and grouped into a set, $Y_i = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_m\}$. All the *C* sets further form the training set, $Y = \{Y_1, Y_2, \cdots, Y_C\}$.

During the testing phase, feature descriptors are extracted for each single-view query image and propagated down the vocabulary tree by the same fashion to obtain a single weighted query histogram $X = \{\boldsymbol{x}\}$. The query image is then given a single-view relevance score s based on the ℓ_1-normalized difference between the weighted query and the ith training set Y_i:

$$s(x, Y_i) = \min_{\boldsymbol{y}_j \in Y_i} \|\frac{\boldsymbol{x}}{\|\boldsymbol{x}\|_1} - \frac{\boldsymbol{y}_j}{\|\boldsymbol{y}_j\|_1}\|_1. \tag{10}$$

When multiple-view histograms of the query object are available, $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_L\}$, a new method to perform joint classification is necessary to take into account the multiple-view information. In this case, the median of the

single-view relevance scores is used to determine the averaged multiple-view relevance score:

$$s(X, Y_i) = \text{median}_{\boldsymbol{x}_j \in X} s(\boldsymbol{x}_j, Y_i). \quad (11)$$

We choose median as a robust mean operator, which is more suitable for situations where some query images are not well matched with any training images in 3-D. Notice that when X only contains a single camera view, (11) is identified as (10).

Finally, the label of the object category for the multiple-view histograms is assigned as:

$$\text{label}(X, Y) = \arg \min_{i \in [1 \cdots C]} s(X, Y_i), \quad (12)$$

which is simply the object category that achieves the minimal multiple-view relevance score.

In this paper, we are concerned with the implementation of the above multiple-view recognition system on a band-limited camera sensor network. As shown in Figure 6, on the sensor side, each query histogram after the quantization process is projected onto a lower-dimensional feature space by random projection, and transferred to a base-station computer. On the computer side, the received feature vectors $\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_L$ are jointly decoded in (4) by ℓ_1-min to obtain the estimates $X = \{\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}_2, \cdots, \hat{\boldsymbol{x}}_L\}$ of the original weighted histograms. Finally, the joint classification algorithm (12) is employed to recover a label of the object that minimizes the multiple-view relevance score s.

5 Experiment

5.1 Setup

We use the BMW database to benchmark the performance of the algorithm (12). First, we divide the database into a training set and a testing set. As the vantage points of each object are named numerically from 0 to 15, images from all the even number locations are designated as the training set, and the ones from the odd number locations are assigned to the testing set. Furthermore, since the main purpose of the experiment is to validate the recognition performance of using multiple-view *testing* images, we do not include the redundant multiple views in the training set. More specifically, only training images from a single camera, i.e., Cam 2, are used for the construction of the vocabulary tree and for the subsequent recognition process.

Based on the BMW database, we choose to compare how discriminative three existing robust feature descriptors are in representing the image appearance of objects, namely, SIFT [13], SURF [2], and CHoG [5]. The original SIFT framework includes a gradient-based interest-point detector with a single-scale 128-D descriptor for each feature. The SURF algorithm is based on sums of approximated 2D Haar wavelet responses, and it also makes use of integral images to speed up the keypoint detection and descriptor extraction. The quantization process yields a 64-D vector. The relatively newer CHoG feature detector and descriptor has been specially designed for platforms with low processing capabilities, and yields a 45-D descriptor for each detected feature.

We design two testing scenarios to evaluate the performance of the distributed recognition scheme, namely, the small-baseline and the large-baseline scenarios. In the small-baseline scenario, images captured concurrently from multiple cameras at one vantage point are used to determine the object category. We evaluate the recognition performance using one camera (i.e., Cam 2), two cameras (i.e., Cam 1 and Cam 2), and three cameras (i.e., Cam 1, Cam 2, and Cam 3). In the large-baseline scenario, images captured from one to three vantage points are randomly chosen from the same testing category for recognition. The two scenarios are well illustrated in Figure 2.

In terms of system implementation, the CITRIC mote has been shown to have the capacities to locally extract and compress high-dimensional histograms [24]. Nevertheless, in this paper, the data processing and classification on the BMW database are performed on a Linux workstation. All the code has been implemented in MATLAB/C++ with a MEX compiler interface.

5.2 Small-Baseline Results

To establish a baseline performance, we first evaluate the recognition accuracy of (12) without involving the random-projection and ℓ_1-min codec. In other words, we assume the classifier can directly access and process all the images in their full resolution. Table 1 shows the recognition rates for the three camera configurations based on the SIFT, SURF, and CHoG feature descriptors. It shows that in all the three cases, the recognition rates improve when more views of the query object are included in the global recognition scheme. Overall, CHoG features yield the best recognition rates compared to the other two feature descriptors. We find this to our benefit, as CHoG features have been designed for distributed wireless camera applications [5], and thus have the lowest dimensionality and extraction time compared to SURF and SIFT feature descriptors. For this reason, we will choose the CHoG features exclusively for the multiple-view recognition experiment in the rest of the section.

Table 1: Small-baseline recognition rates without histogram compression. The best rates are marked in bold face.

Expt.	# Train Images	# Test Images	SIFT Rate(%)	SURF Rate(%)	CHoG Rate(%)
1 Cam	160	160	71.25	80.62	**81.88**
2 Cam	160	320	72.5	81.25	**84.38**
3 Cam	160	480	73.75	81.88	**86.25**

Next, we activate the ℓ_1-min codec in the same camera configurations, and evaluate the recognition accuracy when the query histograms are projected from its original 10,000-D space to lower projection dimensions ranging from 1000 to 9000. For each projection dimension d and each camera sensor j, we create a fixed random projection matrix A_{dj} offline. The ℓ_1-min algorithm to reconstruct the JS signals (4)

is based on the alternating direction method [25]. Figure 7 shows the recognition rates for the three experiments against the projected dimension.

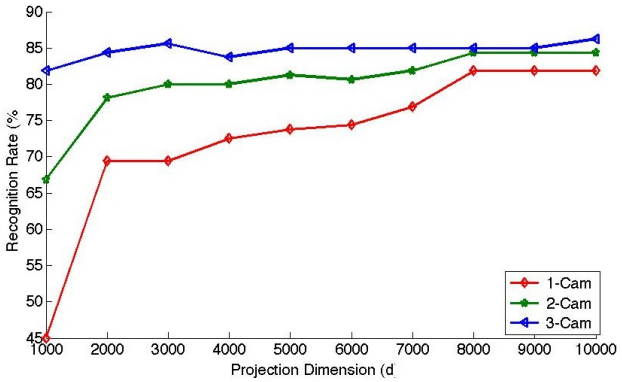

Figure 7: Comparison of the CHOG recognition rates (in color) in the small-baseline scenario with different random projection dimensions.

We observe that, with small projection dimensions close to 1000, the recognition rates using two or three cameras improves significantly compared to the single-view recognition rates. For instance, at $d = 1000$, the recognition rate from a single camera (i.e., Cam 2) is about 45%. The rate is boosted to 68% with two cameras and 82% with three cameras. It is also important to note that the improved recognition rates using the multiple-view information are also higher than merely increasing the projection dimension in the single-camera scenario. For instance, The recognition rate for 2-Cam at $d = 2000$ is higher than the rate for 1-Cam at $d = 4000$.

As the projection dimension increases, the recognition rates for the three scenarios increase as well and reach a plateau beyond $d = 8000$. Interestingly, for the 3-Cam case, the ground-truth recognition rate of 85% is achieved in a very low projection space of 3000-D.

5.3 Large-Baseline Results

The large-baseline performance is evaluated using the same procedure as in the small-baseline experiments. Table 2 shows the recognition rates for the three camera configurations without involving the ℓ_1-min codec. Again, the recognition rates improve when more views of the query object are included in the global recognition scheme. Recognition using the CHOG features not only outperforms that with the other two feature descriptors, but is also drastically better than the CHOG recognition rates in the small-baseline experiments of Section 5.2. Specifically, there is about 10% improvement in the recognition rates in the 3-camera case. The result demonstrates that multiple large-baseline images contain much more information about a common object in 3-D than a set of small-baseline images.

When the ℓ_1-min codec is included, Fig. 8 shows the recognition rates versus the random projection dimension. Clearly, the recognition rates using a single camera does not change from the small-baseline scenario. As shown in the plot, the recognition rates at the low projection dimension of 1000 are lower than those of the small-baseline scenario for the 2 and 3-cam cases. However, as the projection dimension increases, the multiple-view recognition rates reach about 95% and begin to plateau. Such rates are never achieved even without random projection in the single view case.

Table 2: Large-baseline recognition rates without histogram compression. The best rates are marked in bold face.

Expt.	# Train Images	# Test Images	SIFT Rate(%)	SURF Rate(%)	CHoG Rate(%)
1 Cam	160	160	71.25	80.62	**81.88**
2 Cam	160	320	76.88	88.13	**93.75**
3 Cam	160	480	83.13	90.00	**94.88**

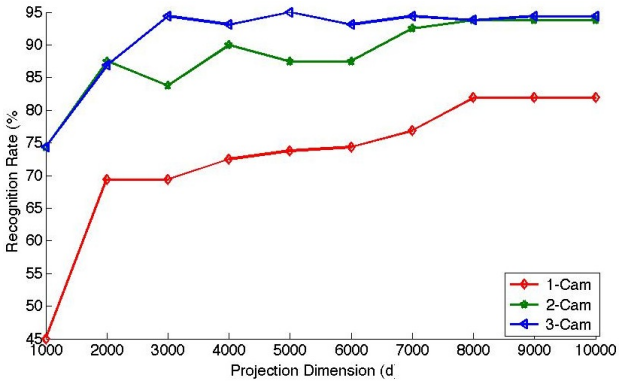

Figure 8: Comparison of the CHOG recognition rates (in color) in the large-baseline scenario with different random projection dimensions.

6 Conclusion and Discussion

We have presented a framework to jointly classify objects observed from multiple vantage points in a distributed wireless camera network. The method is well suited for situations where the camera sensors and the base station are connected only by a band-limited communication channel, and the multiple-view information of the object is available to boost the global recognition. We have drawn from recent developments in compressive sensing theory to formulate a distributed compression scheme to transmit high-dimensional object histograms from camera sensors viewing a common object in 3-D. Most importantly, the algorithm does not require any calibration between the cameras. Therefore, it is very flexible to the addition or omission of some cameras in the network, and the cameras can also be mounted on mobile robot platforms. Finally, we have constructed a new multiple-view object database, namely, the BMW database. The performance of the system has been extensively validated using the database.

Our investigation also has led to several intriguing open problems for future investigation. First, the multiple-view

images may adversely introduce large amounts of outlying features from different background images into the recognition process. However, it is possible to reject these features by considering the geometric consistency between the multiple views during the (offline) training process, such as using the RANSAC technique in [21]. Second, in the paper, the best recognition rate based on the images of the 20 landmarks is about 95%. To successfully deploy such systems in real-world surveillance applications, the recognition rates have to be improved dramatically (e.g., $> 99\%$). Finally, robust techniques must be studied to deal with situations where multiple objects of interest are captured in the images, or certain test images are irrelevant (as outliers) to the given training categories. The new BMW database provides a good public platform to further extend our investigation in these directions.

Acknowledgements

The authors acknowledge Posu Yan of the University of California, Berkeley, for his contribution during the process of image acquisition for the BMW database.

References

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

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

[3] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

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

[5] V. Chandrasekhar, G. Takacs, D. Chen, S. Tsai, R. Grzeszczuk, and B. Girod. CHoG: Compressed histogram of gradients a low bit-rate feature descriptor. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2009.

[6] 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.

[7] 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.

[8] 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.

[9] D. Donoho. For most large underdetermined systems of linear equations the minimal l_1-norm solution is also the sparsest solution. *Communications on Pure and Applied Math*, 59(6):797–829, 2006.

[10] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. In *IEEE CVPR Workshop on Generative-Model based Vision*, 2004.

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

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

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

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

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

[16] 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.

[17] J. Philbin and A. Zisserman. The oxford buildings dataset. `http://www.robots.ox.ac.uk/~vgg/data/oxbuildings/`.

[18] J. Sivic and A. Zisserman Video Google: A text retrieval approach to object matching in videos. In *Proceedings of the IEEE International Conference on Computer Vision*, 2003.

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

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

[21] P. Turcot and D. Lowe. Better matching with fewer features: The selection of useful features in large database recognition problems. In *ICCV Workshop on Emergent Issues in Large Amounts of Visual Data*, 2009.

[22] S. Wright, R. Nowak, and M. Figueiredo. Sparse reconstruction by separable approximation. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2008.

[23] A. Yang, A. Ganesh, S. Sastry, and Y. Ma. Fast ℓ_1-minimization algorithms and an application in robust face recognition: a review. Technical Report UCB/EECS-2010-13, UC Berkeley, 2010.

[24] A. Yang, S. Maji, C. Christoudias, T. Darrell, J. Malik, and S. Sastry. Multiple-view object recognition in band-limited distributed camera networks. In *Proceedings of the ACM/IEEE International Conference on Distributed Smart Cameras*, 2009.

[25] J. Yang and Y. Zhang. Alternating direction algorithms for ℓ_1-problems in compressive sensing. *(preprint) arXiv:0912.1185*, 2009.

[26] 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.