Chapter 1

MULTIPLE-VIEW OBJECT RECOGNITION IN SMART CAMERA NETWORKS

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

Abstract We study object recognition in low-power, low-bandwidth smart 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 overcome visual nuisances such as occlusion and pose variations between multiple camera views. To accommodate limited bandwidth between the cameras and the base-station computer, the method utilizes the available computational power on the smart sensors to locally extract SIFT-type image features to represent individual camera views. We show that between a network of cameras, high-dimensional SIFT histograms exhibit a joint sparse pattern corresponding to a set of shared features in 3-D. Such joint sparse patterns can be explicitly exploited to encode the distributed signal via random projections. At the network station, multiple decoding schemes are studied to simultaneously recover the multiple-view object features based on a 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 experiment.

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

1. Introduction

Object recognition has been extensively studied 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 more known object classes. Successful methods have been demonstrated in the past, including pedestrian detection [16], general object detection[1, 27] (e.g., vehicles and animals), and scene annotation

[17, 24] (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 [20] combined with a descriptor, e.g., SIFT (Scale-Invariant Feature Transform) [18, 3].

In this paper, we consider a relatively new scenario where a network of smart cameras are set up to simultaneously acquire an ensemble of images when a common object can be viewed from multiple vantage points. Traditionally, investigators often assume that the cameras are reliably connected to a central computer with no bandwidth limitation. 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.

Recent studies in distributed object recognition have been mainly focused on two directions. First, when multiple images share a set of common visual features, correspondence can be established across camera views. This indeed was the original motivation of the SIFT framework [18]. More recently, [11, 27] proposed to harness the prior spatial distribution of specific features to guide the multiple-view matching process and improve recognition. Using random projections, [31] 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 bandwidth to streamline the high-dimensional visual features and perform feature matching, distributed data compression [10] can be utilized to encode and transmit the features. In particular, several methods have been proposed to directly compress 2-D visual features, such as PCA, entropy coding, and semantic hashing [14, 26, 19, 29]. For SIFT-type visual histograms, [5] proposed a rate-efficient codec to compress scalable tree structures in describing the hierarchy of histograms. In another work, [8] studied a multiple-view SIFT feature selection algorithm. The authors argued that the number of SIFT features that need to be transmitted can be reduced by considering the joint distribution of the feature histograms among multiple camera views. However, the selection of the joint features depends on learning the mutual information among different camera views, and their relative positions must be fixed.

Contributions

We propose a novel distributed object recognition system suitable for band-limited smart camera networks. The contributions of this paper are two-fold: First, based on compressive sensing theory, we propose an effective distributed

compression scheme to encode high-dimensional *visual 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 compression for multiple-view recognition. No communication between the cameras is necessary to exchange mutual information about the scene. Random projections will be used to provide dimensionality reduction, which is particularly adept for sensor network applications. Note that the paper does not address compression of object images or SIFT-type visual features *per se*, which may not exhibit joint sparse pattens across multiple camera views.

Second, we detail the design of the distributed recognition system. On the sensor side, a smart camera sensor platform called CITRIC [6] is utilized. The substantial computational capability on CITRIC enables a fast implementation of the SURF (Speeded-Up Robust Features) detector [3] and compression of the histograms. On the station side, we demonstrate that the multiple-view object histograms can be jointly recovered using ℓ^1-minimization (ℓ^1-min) solvers. Finally, the object class from the multiple views is classified using standard *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 [21] is used.

2. Encoding Multiple-View Features via Sparse Representation

Suppose multiple camera sensors are equipped to observe a 3-D scene from multiple vantage points. As the wireless network and its placement is not an essential part of the paper, we can assume the sensors communicate with a base station via a single-hop wireless network. Using a visual feature detector (e.g., SIFT or SURF), viewpoint-invariant features can be extracted from the images, as shown in Figure 1.1. These local features are called *codewords*.

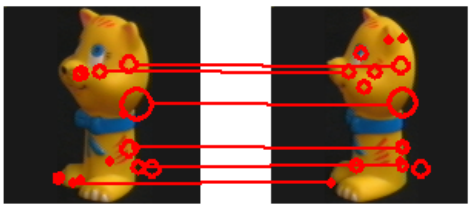

Figure 1.1. Detection of SURF features (red circles) on two image views of a toy. The correspondence of the interest points is highlighted via red lines.

If one is given a large set of training 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 [22]. 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 the visual features in a single object image is then *sparse*, which is called a *feature histogram*. 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 1.2.

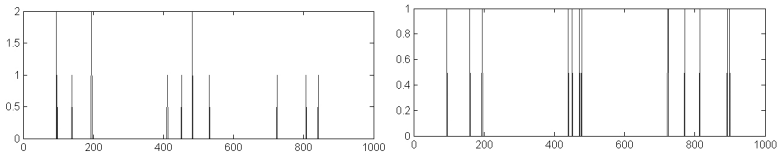

Figure 1.2. The histograms representing the image features in Figure 1.1.

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 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 features in the multiple views. Hence, a* joint sparsity *(JS) model is defined as:*

$$\begin{aligned} \boldsymbol{x}_1 &= \tilde{\boldsymbol{x}} + \boldsymbol{z}_1, \\ &\vdots \\ \boldsymbol{x}_L &= \tilde{\boldsymbol{x}} + \boldsymbol{z}_L. \end{aligned} \tag{1.1}$$

In (1.1)*, $\tilde{\boldsymbol{x}}$ is called the* common sparsity*, and $\boldsymbol{z}_i$ is called an* innovation *[10]. Both $\tilde{\boldsymbol{x}}$ and $\boldsymbol{z}_i$ are also sparse and nonnegative.*

Suppose no communication is allowed between the L cameras:

1. *On each camera, employ 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. *At the base station, once $\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_L$ are received, simultaneously recover the histograms $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_L$ and classify the object class.*

Random Projections

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

$$f : \boldsymbol{y} = A\boldsymbol{x}, \tag{1.2}$$

where $A \in \mathbb{R}^{d \times D}$ is in general a full-rank matrix with $d < D$.

Recently, a special projection function called *random projections* has gained much publicity in applications where the prior information of the data and the computational power of the sensor modalities are limited [4, 6, 31]. In this case, each element a_{ij} of A is independently drawn from a zero-mean Gaussian distribution. One can further simplify the implementation by using a Bernoulli distribution of two values $(+1, -1)$ with equal probability.

Compared with other linear projections, the main advantages of random projections are two-fold: 1. Random projections are efficient to generate using a pseudo-random number generator, and they do 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 (1.2).

Clearly, encoding $\boldsymbol{x}$ using an overcomplete dictionary A in (1.2) leads to infinitely many solutions. Before we discuss how to uniquely recover $\boldsymbol{x}$, remember the goal of Problem 1 is to classify the object class in a 3-D scene. Indeed, one can directly utilize the randomly projected features $\boldsymbol{y}$ in the d-dim feature space for recognition. An important property of random projections is that they preserve the pairwise Euclidean distance, known as the *Johnson-Lindenstrauss* (J-L) lemma [13]:

THEOREM 1.1 (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 $d \geq 4(\epsilon^2/2 - \epsilon^3/3)^{-1} \log n$, random projections $f : \mathbb{R}^D \to \mathbb{R}^d$ preserve the pairwise Euclidean distance with high probability:*

$$(1-\epsilon)\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2 \leq \|f(\boldsymbol{x}_i) - f(\boldsymbol{x}_j)\|_2^2 \leq (1+\epsilon)\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2, \qquad (1.3)$$

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 projections have been applied to reducing the data complexity for k-nearest neighbor (kNN) [2, 31]. On the other hand, note that Gaussian random projections do 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 histogram intersection kernels [12], which will be discussed in more detail later. For clarity, in the paper, our discussion will be limited to Gaussian random projections.

Another observation about 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., in the case of feature histograms computed over a large vocabulary, each $\boldsymbol{x}$ then can be reliably recovered from its random observations $\boldsymbol{y}$. This "inverse" process is the main subject in compressive sensing [4, 9].

THEOREM 1.2 *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* (1.2) *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:*

$$(P_1): \quad \min \|\boldsymbol{x}\|_1 \text{ subject to } \boldsymbol{y} = A\boldsymbol{x}. \tag{1.4}$$

Clearly, the condition ρ in Theorem 1.2 is a function of the matrix A. In fact, for a particular A matrix, ρ can be exactly quantified in convex polytope theory [9]. In the rest of this subsection, we will first overview this relationship.

Figure 1.3 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.

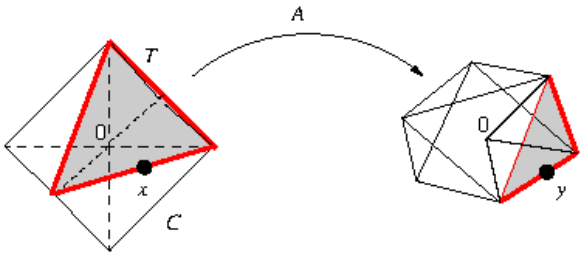

Figure 1.3. Projection of a cross polytope C in $\mathbb{R}^3$ to a quotient polytope AC via projection A. The corresponding simplex is T at the shaded area. Both AC and AT are 0-neighborly.

THEOREM 1.3 *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 1.3 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 1.3, 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 projections: In a high-dimensional space, with high probability, random projections preserve most faces of a cross polytope. A short explanation to this observation is that most randomly generated column vectors in A are linearly independent.

Enforcing Nonnegativity in ℓ^1-Minimization

Given the observations $\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_L$, (P_1) 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 therefore are strictly nonnegative, and it also fails to enforce the possible joint sparse patterns that are shared among multiple camera views. In this subsection, we first discuss how to impose nonnegativity in ℓ^1-min.

Assuming nonnegative $\boldsymbol{x}$ is normalized to have unit ℓ^1-norm without loss of generality, we denote $T \doteq T^{D-1}$ as the standard simplex in $\mathbb{R}^D$, i.e.,

$$T = \{\boldsymbol{x} : \|\boldsymbol{x}\|_1 = 1 \text{ and } \boldsymbol{x} \geq 0\}. \tag{1.5}$$

Figure 1.3 shows the relationship between C^D and T^{D-1}. Hence, a k-sparse nonnegative vector $\boldsymbol{x}$ must lie on a $(k-1)$-face of T, which is only 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 [9]:

THEOREM 1.4 *1 Any nonnegative sparse signal $\boldsymbol{x} \in \mathbb{R}^D$ with k or less sparse coefficients can be recovered by*

$$(P_1'): \quad \min \|\boldsymbol{x}\|_1 \text{ subject to } \boldsymbol{y} = A\boldsymbol{x} \text{ and } \boldsymbol{x} \geq 0 \tag{1.6}$$

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

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

The nonnegative ℓ^1-min (1.6) is a linear program, and can be solved by efficient algorithms, such as *orthogonal matching pursuit* (OMP) and *polytope faces pursuit* (PFP) [23]. These algorithms are usually preferred in sensor network applications compared to other more expensive programs (e.g., interior-point methods [7, 28]). In both simulation and experiment on real-world data, we have found that PFP is a more efficient algorithm than interior-point methods to impose the nonnegativity constraint, and produces better results than OMP. In the rest of the paper, PFP is the ℓ^1-solver of our choice.

Estimation of Joint Sparse Signals

We propose to adopt a *joint sparsity* (JS) model [10] to directly recover the common sparse signal and the sparse innovations as the following:

$$\begin{aligned} \boldsymbol{y}_1 &= A_1(\tilde{\boldsymbol{x}} + \boldsymbol{z}_1) = A_1\tilde{\boldsymbol{x}} + A_1\boldsymbol{z}_1, \\ &\vdots \\ \boldsymbol{y}_L &= A_L(\tilde{\boldsymbol{x}} + \boldsymbol{z}_L) = A_L\tilde{\boldsymbol{x}} + A_L\boldsymbol{z}_L, \end{aligned} \tag{1.7}$$

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

$$\begin{aligned} \begin{bmatrix} \boldsymbol{y}_1 \\ \vdots \\ \boldsymbol{y}_L \end{bmatrix} &= \begin{bmatrix} A_1 & A_1 & 0 & \cdots & 0 \\ \vdots & & \ddots & \ddots & \\ A_L & 0 & \cdots & 0 & A_L \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{x}} \\ \boldsymbol{z}_1 \\ \vdots \\ \boldsymbol{z}_L \end{bmatrix} \\ \Leftrightarrow \qquad \boldsymbol{y}' &= A'\boldsymbol{x}' \in \mathbb{R}^{dL}. \end{aligned} \tag{1.8}$$

The global projection function (1.8) projects a $D(L+1)$-dim nonnegative sparse signal $\boldsymbol{x}'$ 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 (1.2), $\rho = \frac{k}{d}$, and $\|\tilde{\boldsymbol{x}}\|_0 = \frac{k}{2}$ and $\|\boldsymbol{z}_i\|_0 = \frac{k}{2}$. Then the new sparsity ratio in (1.8) becomes

$$\rho' = \frac{(L+1)k/2}{dL} = \frac{L+1}{2L}\rho. \tag{1.9}$$

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

EXAMPLE 1.5 *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 common sparsity and innovation are constructed, as shown in Table 1.1, 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, and JS, 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 1.1. Average ℓ^0-error and ℓ^2-error of OMP, PFP, and JS over 100 trials. The two numbers in the parentheses indicate the sparsity in the common 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_{JS}	**1.85**	**1.65**	**1.95**
ℓ^2_{JS}	**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.*

Overall, the JS model achieves the best performance. First, w.r.t. different combinations of common 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 in the 1000-D ambient space.*

Finally, taking advantage of the JS model, flexible strategies can be proposed for choosing the random projection dimensions d_i. A *necessary* condition for simultaneously recovering $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_L$ can be found in [10]. Basically, it requires each sampling rate $\delta_i = d_i/D$ guarantees that the so-called *minimal sparsity signal* of $\boldsymbol{z}_i$ is sufficiently encoded, and the total sampling rate must also guarantee that both the common sparsity and the innovations are sufficiently encoded. [1] This result suggests a flexible strategy for choosing varying sampling rates and communication bandwidth, that is, the random project dimensions d_i need *not* to be the same for the L sensors to guarantee perfect recovery of the distributed data. For example, sensor nodes in a network that have lower bandwidth or lower power reserve can choose to reduce the sampling rate in order to preserve energy.

3. System Implementation

The complete recognition system has been implemented on the Berkeley CITRIC smart camera platform [6] and a computer as the base station. The design of the CITRIC platform provides considerable computational power to execute SURF 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 SURF features.[2] The 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. Figure 1.4 illustrates two examples. 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.

Figure 1.4. SURF features detected from a corridor scene (left) and a tree object (right).

To measure the speed of the system on the camera sensor, we have conducted a real-world experiment at multiple locations in an office building [30]. 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 random projections. We believe the limitation of the CITRIC platform for computation-intensive applications 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 TelosB network mote.

At 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. 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[3] with the histogram intersection kernel for learning classifiers. This kernel and its variants such as the pyramid match kernel [12] have been shown to work quite well for visual recognition.

When multiple views are available, there are various ways to use them 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 a 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.

4. 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-SURF descriptors 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 [15]. We used 4 levels and a branching factor of 10 to get a 991 word vocabulary at the finest level of the hierarchy. We represent each image and perform ℓ^1 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.

For each object, 10 training examples are sampled uniformly from the complete 360 degree viewing circle. To classify a query image, we use the projected features from its neighboring views in the dataset to jointly recover the features, but the classification is performed on a per-view basis for a fair comparison of the quality of the distributed compression algorithm.

Figure 1.5 shows the performance of various compression 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 SURF histograms. Hence, the upper-bound per-view recognition rate is about 95%. When the histograms are compressed via random projections, 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 sparse histograms in the high-dimensional space, when the dimension of random projections 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 significantly boosts the accuracy by as much as 50%, as seen in Figure 1.5. 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. JS has also been shown to reduce the ℓ^1- and ℓ^2-recovery-error in Figure 1.6.

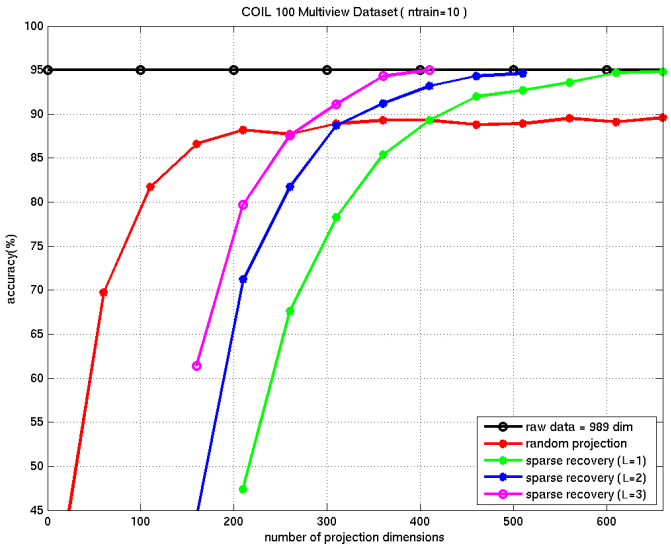

Figure 1.5. Per-view classification accuracy (in color) w.r.t. random projection dimensions: 1. Randomly projected features. 2. PFP in one view. 3. JS in two to three views. The top curve is the baseline recognition rate without histogram compression.

5. 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, nonnegativity, and multiple-view joint sparsity. The complete recognition system has been implemented on the Berkeley CITRIC smart camera platform.

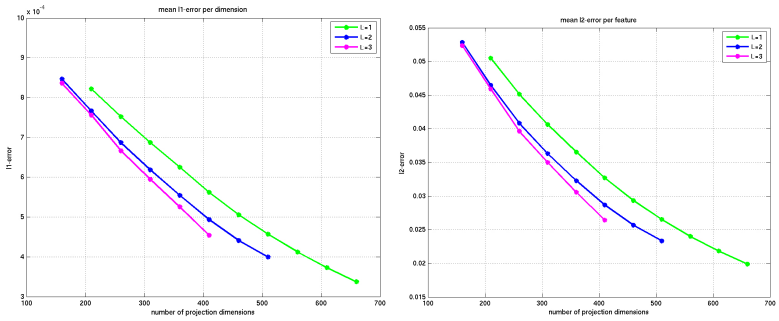

Figure 1.6. (Left) ℓ^1-error and (Right) ℓ^2-error using 1–3 camera views.

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 visual features can improve the classification across multiple camera views in a band-limited sensor network.

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 SURF on the CITRIC camera platform.

Notes

1. The strategy of choosing varying sampling rate is a direct application of the celebrated Slepian-Wolf theorem [25]. In a nutshell, the theorem shows that, given two sources X_1 and X_2 that generate sequences x_1 and x_2, asymptotically, the sequences can be jointly recovered with vanishing error probability *if and only if*

$$R_1 > H(X_1|X_2), \quad R_2 > H(X_2|X_1), \quad R_1 + R_2 > H(X_1, X_2),$$

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.

2. The Open SURF project is documented at: `http://code.google.com/p/opensurf1/`.
3. `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`.

References

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

[2] Ailon, N. and Chazelle, B. (2006). "Approximate nearest neighbors and the fast Johnson-Lindenstrauss transform," in Proceedings of the ACM Symposium on Theory of Computing, 2006.

[3] Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. (2008). "SURF: Speeded Up Robust Features," Computer Vision and Image Understanding.

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

[5] Chen, D., et al. (2009). "Tree histogram coding for mobile image matching," in Proceedings of the IEEE Data Compression Conference.

[6] Chen, P., et al. (2008). "CITRIC: A low-bandwidth wireless camera network platform," in Proceedings of the International Conference on Distributed Smart Cameras.

[7] Chen, S., Donoho, D., and Saunders, M. (2001). "Atomic decomposition by basis pursuit," SIAM Review.

[8] Christoudias, C., Urtasun, R., and Darrell, T. (2008). "Unsupervised feature selection via distributed coding for multi-view object recognition," in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition.

[9] Donoho, D. and Tanner, J. (2005). "Neighborliness of randomly projected simplices in high dimensions," PNAS.

[10] Duarte, M., et al. (2005). "Distributed compressed sensing of jointly sparse signals," in Proceedings of the Thirty-Ninth Asilomar Conference on Signals, Systems and Computers.

[11] Ferrari, V., Tuytelaars, T., and Van Gool, L. (2004). "Integrating multiple model views for object recognition," in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition.

[12] Grauman, K. and Darrell, T. (2005). "The pyramid match kernel: discriminative classification with sets of image features," in Proceeding of International Conference on Computer Vision.

[13] Johnson, W. and Lindenstrauss, J. (1984). "Extensions of Lipschitz maps into a Hilbert space," Contemporary Mathematics.

[14] Ke, Y. and Sukthankar, R. (2004). "PCA-SIFT: A more distinctive representation for local image descriptors," in Proceedings of International Conference on Computer Vision and Pattern Recognition.

[15] Lee, J. (2008). "A pyramid match toolkit," MIT CSAIL Tech Report: MUT-CSAIL-TR-2008-017.

[16] Leibe, B., Seemann, E., and Schiele, B. (2005). "Pedestrian detection in crowded scenes," in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition.

[17] Liu, C., et al. (2008). "SIFT flow: dense correspondence across different scenes," in Proceedings of European Conference on Computer Vision.

[18] Lowe, D. (1999). "Object recognition from local scale-invariant features," in Proceedings of International Conference on Computer Vision.

[19] Makar, M., Chang, C., Chen, D., Tsai, S., and Girod, B. (2009). "Compression on image patches for local feature extraction," in Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing.

[20] Mikolajczyk, K., et al. (2005). "A comparison of affine region detectors," International Journal of Computer Vision.

[21] Nene, S., Nayar, S., and Murase, H. (1996). "Columbia object image library (COIL-100)," Columbia University Tech Report: CUCS-006-96.

[22] Nistér, D. and Stewénius, H. (2006). "Scalable recognition with a vocabulary tree," in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition.

[23] Plumbley, M. (2006). "Recovery of sparse representations by polytope faces pursuit," in Proceedings of International Conference on Independent Component Analysis and Blind Source Separation.

[24] Quattoni, A., Collins, M., and Darrell, T. (2008). "Transfer learning for image classification with sparse prototype representations," in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition.

[25] Slepian, D. and Wolf, J. (1973). "Noiseless coding of correlated information sources," IEEE Transactions on Information Theory.

[26] Takacs, G., Chandrasekhar, V., Gelfand, N., Xiong, Y., Chen, W., Bismpigiannis, T., Grzeszczuk, R., Pulli, K., and Girod, B. (2008). "Outdoor augmented reality on mobile phone using loxel-based visual feature organization," in Proceedings of International Multimedia Conference.

[27] Thomas, A., et al. (2006). "Towards multi-view object class detection," in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition.

[28] Tibshirani, R. (1996). "Regression shrinkage and selection via the LASSO," Journal of the Royal Statistical Society B.

[29] Weiss, Y., Torralba, A., and Fergus, R. (2008). "Spectral hashing," in Proceedings of Advances in Neural Information Processing Systems.

[30] Yang, A., et al. (2009). "Distributed compression and fusion of nonnegative sparse signals for multiple-view object recognition," in Proceedings of International Conference on Information Fusion.

[31] Yeo, C., Ahammad, P., and Ramchandran, K. (2008). "Rate-efficient visual correspondences using random projections," in Proceedings of International Conference in Image Processing.