# Unsupervised Feature Selection via Distributed Coding for Multi-view Object Recognition

C. Mario Christoudias, Raquel Urtasun and Trevor Darrell
UC Berkeley EECS & ICSI
MIT CSAIL

## Abstract

*Object recognition accuracy can be improved when information from multiple views is integrated, but information in each view can often be highly redundant. We consider the problem of distributed object recognition or indexing from multiple cameras, where the computational power available at each camera sensor is limited and communication between cameras is prohibitively expensive. In this scenario, it is desirable to avoid sending redundant visual features from multiple views. Traditional supervised feature selection approaches are inapplicable as the class label is unknown at each camera. In this paper we propose an unsupervised multi-view feature selection algorithm based on a distributed coding approach. With our method, a Gaussian Process model of the joint view statistics is used at the receiver to obtain a joint encoding of the views* without *directly sharing information across encoders. We demonstrate our approach on recognition and indexing tasks with multi-view image databases and show that our method compares favorably to an independent encoding of the features from each camera.*

## 1. Introduction

Object recognition often benefits from integration of observations at multiple views. However, when multiple camera sensors exist in a bandwidth limited environment it may be impossible to transmit all the visual features in each image. When the task or target class is not known *a priori* there may be no obvious way to decide which features to send from each view. If redundant features are chosen at the expense of informative features, performance can be worse with multiple views than with a single view, given fixed bandwidth.

We consider the problem of how to select which features to send in each view to achieve optimal results at a centralized recognition or indexing module (see Figure 1). An efficient encoding of the streams might be possible in theory if a class label could be inferred at each camera, enabling the use of supervised feature selection techniques to encode and send only those features that are relevant of that class.

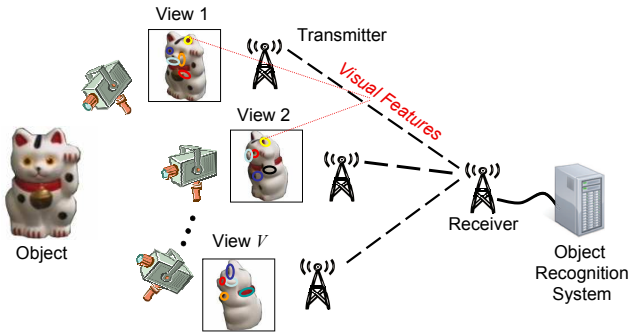


Figure 1. Distributed object recognition. Messages are only sent between each camera (transmitter) and the recognition module (receiver). An efficient joint feature selection is achieved *without* directly sharing information between cameras.

Partial occlusions, unknown camera viewpoint, and limited computational power, however, limit the ability to reliably estimate the image class label at each camera. Instead we propose an unsupervised feature selection algorithm to obtain an efficient encoding of the feature streams.

If each camera sensor had access to the information from all views this could trivially be accomplished by a joint compression algorithm that could, e.g., encode the features of the $v$-th view based on the information in the previous $v-1$ views. We are interested, however, in the case where there is *no* communication between cameras themselves, and messages are only sent from the cameras to the recognition module with a limited backchannel back to the cameras. In practice, many visual category recognition and indexing applications are bandwidth constrained (e.g., wireless surveillance camera networks, mobile robot swarms, mobile phone cameras), and it is infeasible to broadcast images across all cameras or to send the raw signal from each camera to the recognition module.

It is possible to achieve very efficient encoding without any information exchange between the cameras, by adopting a distributed encoding scheme that takes advantage of known statistics of the environment [14, 20, 18, 3]. We develop a new method for distributed encoding based on a Gaussian Process (GP) formulation, and demonstrate its applicability to encoding visual-word feature histograms; such representations are used in many contemporary object

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

indexing and category recognition methods [19, 12, 4].

Our algorithm exploits redundancy between views and learns a statistical model of the dependency between feature streams during an off-line training phase at the receiver. This model is then used along with previously decoded streams to aid feature selection at each camera. If the streams are redundant, then only a few features need to be sent. In this paper, we consider bag-of-words representations [12, 4] and model the dependency between visual feature histograms. As shown in our experiments, our algorithm is able to achieve an efficient joint encoding of the feature histograms without explicitly sharing features across views. This results in an efficient unsupervised feature selection algorithm that improves recognition performance in the presence of limited network bandwidth.

We evaluate our approach using the COIL-100 multi-view image database [11] on the tasks of instance-level retrieval and recognition from multiple views; we compare unsupervised distributed feature selection to independent stream encoding. For a two-view problem, our algorithm achieves a compression factor of over 100:1 in the second view while preserving multi-view recognition and retrieval accuracy. In contrast, independent encoding at the same rate does not improve over single-view performance.

## 2. Related Work

Contemporary methods for object recognition use local feature representations and perform recognition over sets of local features corresponding to each image [8, 12, 4, 22]. Several techniques have been proposed that generalize these methods to include object view-point in addition to appearance [16, 21, 22, 17]. Rothganger et. al. [16] present an approach that builds an explicit 3D model from local affine-invariant image features and uses that model to perform view-point invariant object recognition. Thomas et. al. [21] extend the Implicit Shape Model (ISM) of Leibe and Schiele [8] for single-view object recognition to multiple views by combining the ISM model with the recognition approach of Ferrari et. al. [2]. Similarly, Savarese and Li [17] present a part-based approach for multi-view recognition that jointly models object view-point and appearance.

Traditionally, approaches to multi-view object recognition use only a single input image at test time [16, 21, 22, 17]. Recently, there has been a growing interest in application areas where multiple input views of the object or scene are available. The presence of multiple views can lead to increased recognition performance; however, the transmission of data from multiple cameras places an additional burden on the network. In this paper, we propose an unsupervised feature selection algorithm that enables effective object recognition from multiple cameras in the presence of limited network bandwidth.

Feature selection algorithms exploit data dependency or redundancy to derive compact representations for classification [9]. For our problem, traditional supervised feature selection approaches are inapplicable as the class label is unknown at each camera. Many approaches have been proposed for unsupervised feature selection [1, 9, 13]. Peng et. al. [13] define a minimum-redundancy or maximum-relevance criterion for unsupervised feature selection based on mutual information. Dy and Brodley [1] compute relevant feature subsets using a clustering approach based on a maximum likelihood criterion with the expectation maximization algorithm. For multiple views, it is possible to apply the above unsupervised feature selection techniques independently at each camera to efficiently encode and transmit features over the network. A better encoding of the features, however, can be achieved if features are jointly selected across views [20]. Under limited network bandwidth a joint encoding is not possible as communication between cameras is often prohibitively expensive.

Distributed coding algorithms [14, 20, 18, 3] seek a joint encoding of the streams *without* sharing features across views. These methods exploit data redundancy at a shared, common receiver to perform a distributed feature selection that in many cases approaches the joint encoding rate [20]. Contemporary techniques to distributed coding include the DISCUS algorithm of Pradhan and Ramchandran [14] based on data cosets, and the approach of Schonberg [18] that builds upon low-density parity check codes. In this paper, we present a new distributed coding algorithm for bag-of-words image representations in multi-view object recognition with Gaussian Processes.

Gaussian Processes (GPs) [15] have become popular because they are simple to implement, flexible (i.e., they can capture complex behaviors through a simple parametrization), and are fully probabilistic. The latter enables them to be easily incorporated in more complex systems, and provides an easy way of expressing and evaluating prediction uncertainty. GPs have been suggested as a replacement for supervised neural networks in non-linear regression [15] and they generalize a range of previous techniques (e.g. krigging, splines, RBFs). As shown in our experiments, GPs are well suited for distributed feature selection as the uncertainty measure provided by GPs is correlated with data redundancy. Our work bears similarity to that of Kapoor et. al. [5] that use GP prediction uncertainty as a criteria for example selection in active learning.

## 3. Gaussian Process Review

A Gaussian Process is a collection of random variables, any finite number of which have consistent joint Gaussian distributions [15]. Given a training set $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i), i = 1, \cdots, N\}$, composed of inputs $\mathbf{x}_i$ and noisy outputs $\mathbf{y}_i$, we assume that the noise is additive, independent and Gaussian, such that the relationship between the (latent) function,

$f(\mathbf{x})$, and the observed noisy targets, $\mathbf{y}$, is given by

$$\mathbf{y}_i = f(\mathbf{x}_i) + \epsilon_i \ , \tag{1}$$

where $\epsilon_i \sim \mathcal{N}(0, \sigma^2_{noise})$ and $\sigma^2_{noise}$ is the noise variance.

GP regression is a Bayesian approach that assumes a GP prior over the space of functions,

$$p(\mathbf{f}|\mathbf{X}) = \mathcal{N}(0, \mathbf{K}) \ , \tag{2}$$

where $\mathbf{f} = [f_1, \cdots, f_n]$ is the vector of latent function values, $f_i = f(\mathbf{x}_i)$, $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_N]$, and $\mathbf{K}$ is a covariance matrix whose entries are given by a covariance function, $K_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j)$. GPs are non-parametric models and are entirely defined by their covariance function (and training data); the set of possible covariance functions is defined by the set of Mercer kernels. During training, the model hyperparameters, $\bar{\beta}$, are learned by minimizing

$$-\ln p(\mathbf{X}, \bar{\beta} \,|\, \mathbf{Y}) = \frac{D}{2} \ln |\mathbf{K}| + \frac{1}{2} \mathrm{tr}\left(\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^T\right) + C \ . \tag{3}$$

where $\mathbf{Y} = [\mathbf{y}_1, \cdots, \mathbf{y}_N]$, $C$ is a constant, and $D$ is the dimension of the output.

Inference in the GP model is straightforward, assuming a joint GP prior over training, $\mathbf{f}$, and testing, $\mathbf{f}_*$, latent variables,

$$p(\mathbf{f}, \mathbf{f}_*) = \mathcal{N}\left(0, \begin{pmatrix} \mathbf{K}_{f,f} & \mathbf{K}_{*,f} \\ \mathbf{K}_{f,*} & \mathbf{K}_{*,*} \end{pmatrix}\right) \ , \tag{4}$$

where $*$ is used as shorthand for $f_*$ and the dependency on $\mathbf{X}$ is omitted for clarity of presentation, $\mathbf{K}_{f,f}$ is the covariance of the training data, $\mathbf{K}_{*,*}$, the covariance of the test data, and $\mathbf{K}_{f,*} = \mathbf{K}^T_{*,f}$ is the cross-covariance of training and test data. The joint posterior $p(\mathbf{f}, \mathbf{f}_*|\mathbf{Y})$ is Gaussian:

$$p(\mathbf{f}, \mathbf{f}_*|\mathbf{Y}) = \frac{p(\mathbf{f}, \mathbf{f}_*)p(\mathbf{y}|\mathbf{f})}{p(\mathbf{y})} \ . \tag{5}$$

Marginalizing the training latent variables, $\mathbf{f}$, can be done in closed form and yields a Gaussian predictive distribution [15], $p(\mathbf{f}_*|\mathbf{y}) = \mathcal{N}(\mathbf{M}, \mathbf{C})$, with

$$\mathbf{M} = \mathbf{K}_{*,f}(\mathbf{K}_{f,f} + \sigma^2_{noise}\mathbf{I})^{-1}\mathbf{Y} \tag{6}$$

$$\mathbf{C} = \mathbf{K}_{*,*} - \mathbf{K}_{*,f}(\mathbf{K}_{f,f} + \sigma^2_{noise}\mathbf{I})^{-1}\mathbf{K}_{f,*} \ . \tag{7}$$

The variance of the GP is an indicator of the prediction uncertainty. In the following section we will show how the variance can be used to define a feature selection criteria.

## 4. Distributed Object Recognition

We consider the distributed recognition problem of $V$ cameras transmitting information to a central common receiver with no direct communication between cameras (see Figure 1). In our problem, each camera is equipped with a simple encoder used to compress each signal before transmission. A common decoder receives the encoded signals and performs a joint decoding of the signal streams using a model of the joint statistics. Note that this coding scheme off-loads the computational burden onto the decoder and allows for computationally in-expensive encoders. In what follows, we assume a noiseless channel, but our approach is also applicable to the more general case.

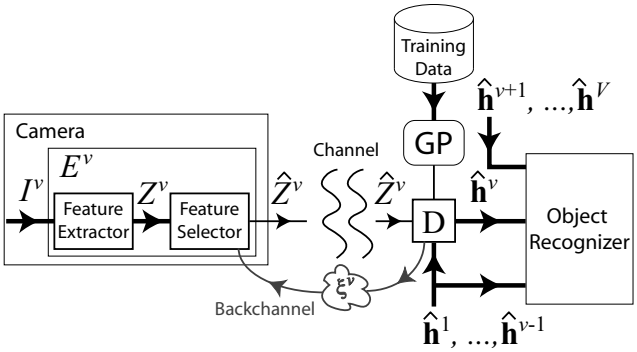


Figure 2. System diagram. Image $I^v$ is coded by encoder $E^v$ and decoder $D$. $\hat{Z}^v$ are the encoded image features, $\hat{\mathbf{h}}^v$ the reconstructed histograms, and $\xi^v$ the non-redundant bin indices for views $v = 1, ..., V$ (see Section 4 for details).

Figure 2 illustrates our proposed distributed coding algorithm at a single camera. With our algorithm, the decoder iteratively queries each of the $V$ cameras and specifies the desired encoding rate the camera should use. At the $v$-th view, the decoder uses its model of joint statistics along with *side information*, i.e., the previously decoded streams, to decode the signal. The use of side information allows the encoder to work at a lower encoding rate than if the stream were encoded independently. As discussed below, the decoder selects the camera encoding rate based on the joint stream statistics and transmits this information back to the encoder. If the $v$-th view is highly redundant with respect to the side information, then little-to-no information needs to be encoded and sent to the decoder.

In this work, we consider bag-of-words models for object recognition [12, 4]. With these models, an image is represented using a set of local image descriptors extracted from the image either at a set of interest point locations (e.g., those computed using a Harris point detector [10]) or on a regular grid. In our experiments, we employ the latter feature detection strategy in favor of simplicity at the encoder. To perform feature coding, the local image features are quantized using a global vocabulary that is shared by the encoder and decoder and computed from training images.

Let $I^v$, $v = 1, ..., V$ be a collection of $V$ views of the object or scene of interest, imaged by each camera and $Z^v$ be the set of quantized local image features corresponding to image $I^v$ computed by the $v$-th encoder, $E^v$. In this context (see Figure 2), the encoders transmit quantized features to the central receiver and the encoding rate is the number

of features sent.

In theory, distributed coding with individual image features (e.g., visual words) might be possible, but preliminary experiments have shown that distributed coding of local features does not improve over independent encoding at each camera. Using a joint model over quantized features on COIL-100 with a 991 word vocabulary gave an entropy of 9.4 bits, which indicates that the joint feature distribution is close to uniform (for a 991 word feature vocabulary, the uniform distribution has an entropy of 10 bits). This is expected since a local image feature is a fairly weak predictor of other features in the image.

We have found, however, that distributed coding of histograms of local features is effective. As seen in our experiments, the distribution over features in one view is predictive of the distribution of features in other views and, therefore, feature histograms are a useful image representation for distributed coding.

### 4.1. Joint Feature Histogram Model

Let $\hat{Z}^v$ be the set of encoded features of each view, $v = 1, ..., V$. To facilitate a joint decoding of the feature streams, the decoder first computes a feature histogram, $\hat{\mathbf{h}}^v = h(\hat{Z}^v)$, using the global feature vocabulary. Note, in our approach, the form of $h(\cdot)$ can either be a flat [19] or hierarchical histogram [12, 4]; we present a general distributed coding approach applicable to any bag-of-words technique. At the decoder, the joint stream statistics are expressed over feature histograms,

$$p(\mathbf{h}^1, \mathbf{h}^2, ..., \mathbf{h}^V) = p(\mathbf{h}^1) \prod_{v=2}^{V} p(\mathbf{h}^v | \mathbf{h}^{v-1}, ..., \mathbf{h}^1), \quad (8)$$

where the conditional probabilities are learned from training data as described in Section 3.

Assuming independence between the histogram bins and pair-wise dependence between histograms we write

$$p(\mathbf{h}^v | \mathbf{h}^{v-1}, ..., \mathbf{h}^1) = \prod_{k=1}^{v-1} \prod_{b=1}^{B} p(h^{v,b} | \mathbf{h}^k) \quad (9)$$

where $h^{v,b}$ is the $b$-th bin of histogram $\mathbf{h}^v$, and $B$ is the number of bins.

The joint model of Equation 8 is used to determine which features at a given camera are redundant with the side information. In particular, redundant features are those that are associated with the redundant bins of the histogram of the current view. Since we are ultimately interested in the feature histograms for performing recognition, the encoders can send either histogram bin counts or the quantized visual features themselves.

We obtain a reconstruction of the feature histogram of each view from the view's encoded features and its side information. Let $\mathbf{h}^v$ be the histogram of interest and $\mathbf{h}^k$, $k = 1, ..., v-1$, its side information, where $v$ is the current view considered by the decoder. From Equation 9 the probability of a histogram $\mathbf{h}^v$ given its side information is found as,

$$p(\mathbf{h}^v | \mathbf{h}^{v-1}, ..., \mathbf{h}^1) = \prod_{k=1}^{v-1} p(\mathbf{h}^v | \mathbf{h}^k) \quad (10)$$

We model the above conditional probability using a GP prior over feature histograms. To make learning more tractable we assume independence between histogram bins

$$p(\mathbf{h}^v | \mathbf{h}^{v-1}, ..., \mathbf{h}^1) = \prod_{b=1}^{B} \mathcal{N}(0, \mathbf{K}^{v-1,b}) \quad (11)$$

where a GP is defined over each bin with kernel matrix $\mathbf{K}^{v-1,b}$. We compute $\mathbf{K}^{v-1,b}$ with a covariance function defined using an exponential kernel over the side information,

$$k^{v,b}(\mathbf{h}_i, \mathbf{h}_j) = \prod_{r=1}^{v} \gamma_b^{-v} \exp\left(\frac{d(\mathbf{h}_i^r, \mathbf{h}_j^r)^2}{\alpha_b^2}\right) + \eta_b \delta_{ij} \quad (12)$$

where $\hat{\mathbf{h}}_i = (\hat{\mathbf{h}}_i^1, ..., \hat{\mathbf{h}}_i^v)$ and $\hat{\mathbf{h}}_j = (\hat{\mathbf{h}}_j^1, ..., \hat{\mathbf{h}}_j^v)$ are multi-view histogram instances, $\gamma_b, \alpha_b$ are the kernel hyper-parameters of bin $b$, which we assume to be the same across

**Algorithm 1** GP Distributed Feature Selection

Let $E^v$ be an encoder, $\xi^v$ be defined over sets of feature histogram bin indices, $\hat{Z}^v$ be defined over sets of encoded features, $\mathbf{H}^v$ be a $N \times B$ matrix of $N$ training examples, $v = 1, ..., V$, and $R_{\max}$ be the desired encoding rate.

$\hat{\mathbf{h}} = \emptyset$
$\xi^1 = \{1, ..., B\}$
**for** $v = 1, ..., V$ **do**
  $\hat{Z}^v = \texttt{request}(E^v, \xi^v)$
  **for** $b = 1, ..., B$ **do**
    **if** $b \in \xi^v$ **then**
      $\hat{h}^{v,b} = \psi(\hat{Z}^{v,b})$
    **else**
      $\hat{h}^{v,b} = (\mathbf{k}_*^{v-1,b})^T (\mathbf{K}^{v-1,b})^{-1} \mathbf{H}^{v,b}$
    **end if**
  **end for**
  $\hat{\mathbf{h}} = (\hat{\mathbf{h}}, \hat{\mathbf{h}}^v)$
  **if** $v < V$ **then**
    **for** $b = 1, ..., B$ **do**
      $\sigma^{v+1,b} = (k^{v,b}(\hat{\mathbf{h}}, \hat{\mathbf{h}}) - (\mathbf{k}_*^{v,b})^{\mathrm{T}} (\mathbf{K}^{v,b})^{-1} \mathbf{k}_*^{v,b})^{\frac{1}{2}}$
    **end for**
    $\xi^{v+1} = \texttt{select}(\boldsymbol{\sigma}^{v+1}, R_{\max})$
  **end if**
**end for**

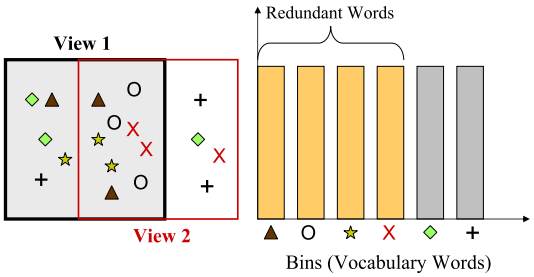


Figure 3. Synthetic example considered below. This scenario consists of two overlapping views of an object, which is presumed to fill the scene. Image features are represented using a 6 word vocabulary.

views, and $\eta_b$ is a per-bin additive noise term. Given training data $\mathbf{H}^v$, where $\mathbf{H}^v$ is a $N \times B$ matrix of $N$ training examples for the $v = 1, ..., V$ views, the kernel hyper-parameters are learned as described in Section 3. We define a different set of kernel hyper-parameters per bin since each bin can exhibit drastically different behavior with respect to the side information.

The variance of each GP can be used to determine whether a bin is redundant: a small bin variance indicates that the GP model is confident in its prediction, and therefore the features corresponding to that bin are likely to be redundant with respect to the side information. In our experiments, we found that redundant bins generally exhibit variances that are small and similar in value and that these variances are much smaller than those of non-redundant bins.

### 4.2. GP Distributed Feature Selection

Distributed feature selection is performed by the decoder using an iterative process. The decoder begins by querying the first encoder to send all of its features, since in the absence of any side information no feature is redundant. At the $v$-th view, the decoder requests only those features corresponding to the non-redundant histogram bins of that view, whose indices are found using the bin variances output by each GP. At each iteration, the GPs are evaluated using the reconstructed histograms of previous iterations as illustrated in Algorithm 1.

Given the encoded features $\hat{Z}^v$, the decoder reconstructs histograms $\hat{\mathbf{h}}^v$, $v = 1, ..., V$, such that bins that are non-redundant are those received and the redundant bins are estimated from the GP mean prediction

$$\hat{h}^{v,b} = \begin{cases} h(\hat{Z}^{v,b}), & b \in \xi^v \\ (k_*^{v-1,b})^{\mathrm{T}}(\mathbf{K}^{v-1,b})^{-1}\mathbf{H}^{v,b}, & \text{otherwise.} \end{cases} \quad (13)$$

where $\mathbf{H}^{v,b} = (h_1^{v,b}, ..., h_N^{v,b})^T$ are the bin values for view $v$ and bin $b$ in the training data, and $\xi^v$ are the bin indices of the non-redundant bins of the histogram of view $v$.

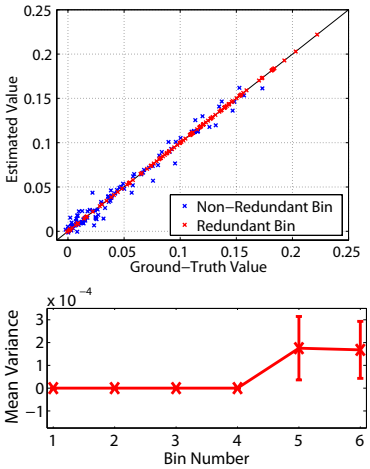


Figure 4. GP variance is correlated with bin redundancy. The GP mean prediction for the second view is plotted vs. ground-truth values for both a redundant and non-redundant bin. The GP variance for each of the 6 histogram bins, averaged across examples is also shown; error bars indicate $\pm 1$ std. deviation. The variance of non-redundant bins is noticeably higher than that of redundant bins.

The GP distributed feature selection algorithm achieves a compression rate proportional to the number of bin indices requested for each view. For view $v$ the compression rate of our algorithm in percent bins transmitted is

$$R = \frac{r}{B} = \frac{2|\xi^v|}{B}, \quad (14)$$

where $B$ is the total number of histogram bins and $r$ is the number of bins received, which is proportional to twice the number of non-redundant bins as a result of the decoder `request` operation. Note, however, that in the case of large amounts of redundancy there are few non-redundant bins encoded at each view and therefore a small encoding rate is achieved.

As mentioned above the bin indices $\xi^v$ are chosen using the GP prediction uncertainty. If a desired encoding rate $R_{\max}$ is provided, the decoder requests the $r_{\max}/2$ histogram bins associated with the highest GP prediction uncertainty (see Equation 14). If $R_{\max}$ is not known, the encoding rate can be automatically determined by grouping the histogram bins at each view into two groups corresponding to regions of high and low uncertainty; $\xi^v$ is then defined using the bins associated with the high uncertainty group. Both strategies exploit the property that prediction uncertainty is correlated with bin redundancy to request the non-redundant bins at each view. Many grouping algorithms are applicable for the latter approach, e.g., conventional clustering. In practice, we use a simple step detection technique to form each group by sorting the bin variances and finding the maximum local difference.

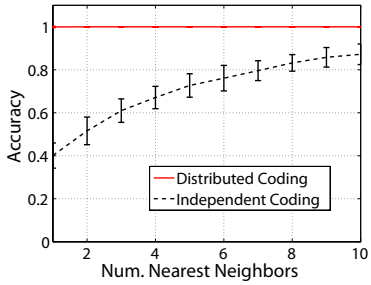


Figure 5. Nearest-neighbor instance-level retrieval for the two-view synthetic dataset; average retrieval accuracy is plotted over varying neighborhood sizes. For a fixed rate, our algorithm far outperforms the independent encoding baseline (see text for details).

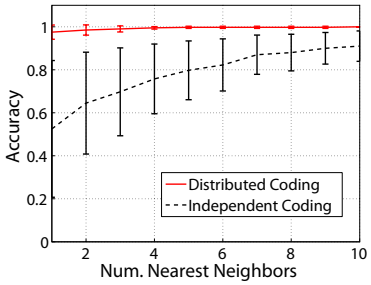


Figure 6. Nearest-neighbor instance-level retrieval on the two-view synthetic dataset with partial redundancy, plotted over varying neighborhood sizes. Our distributed coding algorithm performs favorably to independent encoding even when the bins are only partially redundant.

## 5. Experiments

We evaluate our distributed coding approach on the tasks of object recognition and indexing from multiple views. Given $\mathbf{h}^v$, $v = 1, .., V$, multi-view recognition is performed using a nearest-neighbor classifier over a fused distance measure, computed as the average distance across views

$$D_{i,j}(\mathbf{h}_i, \mathbf{h}_j) = \frac{1}{V} \sum_{v=1}^{V} d(\mathbf{h}_i^v, \mathbf{h}_j^v) \tag{15}$$

where for flat histograms we define $d(\cdot)$ using the $L_2$ norm, and with pyramid match similarity [4] for multi-resolution histograms[1].

We use a majority vote performance metric for nearest-neighbor recognition. Under this metric a query example is correctly recognized if a majority ($\geq k/2$) of its $k$ nearest-neighbors are of the same category or instance. We also experiment with an at-least-one criterion to evaluate performance in an interactive retrieval setting: with this scheme an example is correctly retrieved if one of the first $k$ examples has the true label. We compare distributed coding to independent encoding at each view with a random feature selector that randomly selects histogram bins according to a uniform distribution, and report feature selection performance in terms of percent bins encoded, $R$ (see Equation 14).

In what follows, we first present experiments on a synthetic example with our approach and then discuss our results on COIL-100.

### 5.1. Synthetic Example

To demonstrate our distributed feature selection approach we consider the scenario illustrated in Figure 3. An object is imaged with two overlapping views, and the histograms of each view are represented using a 6 word vocabulary. As shown by the figure, the images are redundant in 4 of the 6 words, as 2 of the words (i.e., diamond and plus) do not appear in the overlapping portion of each view. Although real-world problems are much more complex than described above, we use this simple scenario to give intuition and motivate our approach.

We first consider the case where there is no noise between the redundant features in each view and the redundant features appear only in the overlapping region. We randomly generated $N = 100$, 6-D histograms, where each histogram was generated by sampling its bins from a uniform distribution between 0 and 1, and the histograms were normalized to sum to one. Each histogram was split into two views by replicating the first 4 bins in each view and randomly splitting the other two bins. The above data was used to form a training set of examples, where each pair of histograms corresponds to a single object instance. To form the test set, zero mean Gaussian noise was added to the training set with $\sigma = 0.01$ and the test set histograms were split into two views using the same split ratios as the training set.

For distributed coding we trained 6 GPs, one per dimension, using each view. Figure 4 displays the predicted bin value vs. ground truth for 2 of the bins (one redundant and the other non-redundant) evaluated on the second view of the test set. The GPs are able to learn the deterministic mapping that relates the redundant bins. For the non-redundant bin, the variance of the GP's predictions is quite large compared to that of the redundant bin. Also shown in Figure 4, are the mean GP variances plotted for each histogram bin. The error bars in the plot indicate the standard deviation. The GP variance is much larger for the non-redundant bins than those of the redundant ones whose variances are small and centered about 0. This is expected since non-redundant bins are less correlated and therefore the GPs are less certain in their prediction of the value of these bins from side information.

Evaluating our distributed coding algorithm on the above problem gave a bin rate of $R = 0.66$ in the second view. Figure 5 displays the result of nearest-neighbor instance-

[1]Note our distributed coding algorithm is independent of the choice of classification method.

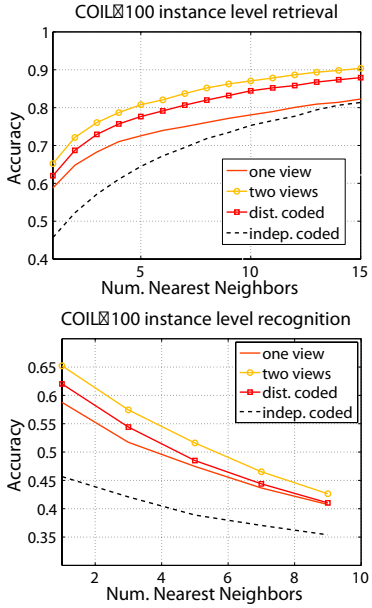


Figure 7. Nearest-neighbor (top) retrieval and (bottom) recognition with two-views on COIL-100. Our algorithm performs significantly better over single view performance under each task while achieving a very low encoding rate. For the retrieval task, our approach performs near multi-view performance. The independent encoding baseline is also shown, where independent feature selection was performed at the same rate as our algorithm. Note that independent encoding with two views does worse than a single view when operating at such a low encoding rate.

level retrieval over each of the 100 instances in the training set for varying neighborhood sizes. The average retrieval accuracy, averaged over 10 independent trials, is shown for both distributed and independent coding of the second view, where for independent coding features were selected at the same rate as distributed coding. Distributed coding far outperforms independent encoding in the above scenario.

We also considered the case of partial redundancy, where the redundant bins are only partially correlated as a result of noise. To simulate partial redundancy we added zero mean Gaussian noise to the split ratios of the first 4 bins with $\sigma = \{0, 0.01, 0.05, 0.1\}$. Figure 6 displays the result of nearest-neighbor recognition with distributed and independent coding of the second view. In the plot, recognition performance is reported, averaged across the different $\sigma$ values, along with error bars indicating the standard deviation. For this experiment, an average bin rate of $R = 0.78 \pm 0.23$ was achieved with our distributed feature selection algorithm. Our distributed coding algorithm can perform favorably to independent encoding even when the bins are only partially redundant.

## 5.2. COIL-100 Experiments

We evaluated our distributed feature selection algorithm using the COIL-100 multi-view object database [11] that consists of 72 views of 100 objects viewed from 0 to 360 degrees in 5 degree increments. A local feature representation is computed for each image using 10 dimensional PCA-SIFT features [6] extracted on a regular grid using a 4 pixel spacing. We evaluate our distributed coding algorithm and perform recognition with the COIL-100 dataset using multi-resolution vocabulary-guided histograms [4] computed with LIBPMK [7]. We split the COIL-100 dataset into train and test sets by taking alternating views of each object. We then paired images 50 degrees apart to form the two views of our problem.

Using the training image features we perform hierarchical $k$-means clustering to compute the vocabulary used to form the multi-resolution pyramid representation. Using 4 levels and a tree branch factor of 10 gave a 991 word vocabulary at the finest level of the hierarchy. GP distributed feature selection is performed over the finest level of the histogram, such that the encoders and decoder only communicate bins at this level. The upper levels of the tree are then recomputed from the bottom level when performing recognition. To perform GP distributed coding we used a kernel defined using L2 distance over a coarse, flat histogram representation.

Figure 7 displays nearest-neighbor retrieval and recognition accuracy using one and two views. A significant performance increase is achieved by using the second view when there are no bandwidth constraints. Applying GP distributed feature selection on the above dataset resulted in a bin rate of $R < 0.01$ in the second view; this is a compression rate of over 100:1. Figure 7 displays the performance of our GP distributed feature selection algorithm. By exploiting feature redundancy across views, our algorithm is able to perform significantly better than single view performance while achieving a very low encoding rate. The result of independent encoding is also shown in the Figure, where independent feature selection was performed at the same rate as our algorithm. In contrast to our approach, independent encoding is not able to improve over single-view performance and in fact does worse at such low encoding rates.

We also tested our approach over different encoding rates, where the desired rate is provided as input to the algorithm. Figure 8 displays the nearest-neighbor performance of our approach over different encoding rates. As expected, nearest-neighbor performance increases for larger encoding rates. Performance saturates at about $r = 50$ bins and remains fairly constant for larger rates. Of coarse, for $r = B$ one would expect to recover ground-truth performance. The slow convergence rate of our approach to ground-truth performance with increasing encoding rate suggests the need for better bin selection criteria, which we plan to investigate as part of future work. The independent encoding baseline is also shown. Recall that at rate $R$ the baseline approach transmits twice the number of bins as our approach

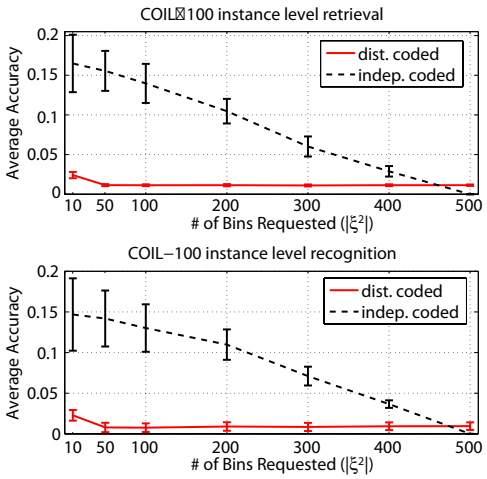


Figure 8. Nearest-neighbor performance increases with encoding rate. Nearest-neighbor performance is shown for the tasks of (top) retrieval and (bottom) recognition. The accuracy difference between our approach and ground-truth two-view performance is shown averaged over neighborhood size; error bars indicate $\pm 1$ std. deviation. The independent encoding baseline is also shown.

as a result of the request operation. Independent encoding needs to transmit nearly the entire histogram ($|\xi^2| = 400$) before reaching a recognition performance close to our approach. Our approach achieves similar performance with only $|\xi^2| = 10$.

## 6. Conclusion and Future Work

In this paper we presented a distributed coding method for unsupervised distributed feature selection and showed its application to multi-view object recognition. We developed a new algorithm for distributed coding with Gaussian Processes and demonstrated its effectiveness for encoding visual word feature histograms on both synthetic and real-world datasets. For a two-view problem with COIL-100, our algorithm was able to achieve a compression rate of over 100:1 in the second view, while significantly increasing accuracy over single-view performance. At the same coding rate, independent encoding was unable to improve over recognition with a single-view. For future work, we plan to investigate techniques for modeling more complex dependencies as well as one-to-many mappings between views and evaluate our approach under different bin selection criteria.

## Acknowledgements

We gratefully acknowledge John Lee for his LIBPMK library [7] and help with our experiments.

## References

[1] J. Dy and C. Brodley. Feature subset selection and order identification for unsupervised learning. In *ICML*, 2000.

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

[3] B. Girod, A. M. Aaron, S. Rane, and D. Rebollo-Monedero. Distributed video coding. *Proceedings of the IEEE*, Jan. 2005.

[4] K. Grauman and T. Darrell. The pyramid match kernel: Efficient learning with sets of features. *JMLR*, 2007.

[5] A. Kapoor, K. Grauman, R. Urtasun, and T. Darrell. Active learning with gaussian processes for object categorization. In *ICCV*, 2007.

[6] Y. Ke and R. Sukthankar. Pca-sift: a more distinctive representation for local image descriptors. *CVPR*, 2004.

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

[8] B. Leibe and B. Schiele. Scale-invariant object categorization using a scale-adaptive mean-shift search. In *DAGM*, 2004.

[9] H. Liu and L. Yu. Feature selection for data mining. Technical report, Arizona State University, 2002.

[10] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool. A comparison of affine region detectors. *IJCV*, 2005.

[11] S. A. Nene, S. K. Nayar, and H. Murase. Columbia object image library (coil-100). Technical report, Colmubia University, 1996.

[12] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *CVPR*, 2006.

[13] H. Peng, F. Long, and C. Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *PAMI*, 2005.

[14] S. S. Pradhan and K. Ramchandran. Distributed source coding using syndromes (discus): design and construction. *Transactions on Information Theory*, 2003.

[15] C. E. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[16] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce. 3d object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints. *IJCV*, 2006.

[17] S. Savarese and L. Fei-Fei. 3D generic object categorization, localization and pose estimation. In *ICCV*, 2007.

[18] D. Schonberg. *Practical Distributed Source Coding and Its Application to the Compression of Encrypted Data*. PhD thesis, University of California, Berkeley, 2007.

[19] J. Sivic and A. Zisserman. *Toward Category-Level Object Recognition*, chapter Video Google: Efficient Visual Search of Videos. Springer, 2006.

[20] D. Slepian and J. Wolf. Noiseless coding of correlated information sources. *Transactions on Information Theory*, 1973.

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

[22] A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing visual features for multiclass and multiview object detection. *PAMI*, 2007.