Fast Sparse Representation with Prototypes

Jia-Bin Huang and Ming-Hsuan Yang
University of California at Merced
{jbhuang,mhyang}@ieee.org

Abstract

Sparse representation has found applications in numerous domains and recent developments have been focused on the convex relaxation of the ℓ_0-norm minimization for sparse coding (i.e., the ℓ_1-norm minimization). Nevertheless, the time and space complexities of these algorithms remain significantly high for large-scale problems. As signals in most problems can be modeled by a small set of prototypes, we propose an algorithm that exploits this property and show that the ℓ_1-norm minimization problem can be reduced to a much smaller problem, thereby gaining significant speed-ups with much less memory requirements. Experimental results demonstrate that our algorithm is able to achieve double-digit gain in speed with much less memory requirement than the state-of-the-art algorithms.

1. Introduction

Recent years have witnessed a considerable resurgence of interest in sparse representation [27, 21, 5]. Much of its popularity as well as effectiveness come from the fact that signals in most problems are structured and can be well represented by a small set of basis vectors. It plays an important role in the success of recent developments in dictionary learning [2, 5] and compressive sensing [8, 9, 13], among others. Given a set of basis vectors (i.e., a dictionary), finding a sparse representation of a signal is often posed as an optimization problem with either ℓ_0-norm or ℓ_1-norm, which usually results in solving an underdetermined linear system. Each sample is then represented as a sparse linear combination of the basis vectors. The complexity of solving ℓ_0-norm minimization problems is known to be NP-hard and numerically unstable. Greedy algorithms such as matching pursuit [25], and orthogonal matching pursuit (OMP) [10, 29] have been proposed to approximate the ℓ_0-norm solution. Although these methods are rather simple and efficient, the solutions are sub-optimal. Recent developments in sparse coding have shown that, under certain assumptions, the solution of ℓ_0-norm minimization problem is equivalent to ℓ_1-norm minimization problem which can be solved by convex optimization [11, 13]. Numerous algorithms have been proposed for ℓ_1-regularized sparse coding [7, 8, 9, 13, 12, 32]. As these algorithms often recast the original problem as a convex program with quadratic constraints, the computational cost for practical applications can be prohibitively high.

For numerous problems in computer vision, machine learning, signal processing, and computer graphics, one simple yet effective approach is to assume that the samples of the same class can be modeled with prototypes or exemplars. Such prototypes can be either the samples themselves, or learned from a set of samples (e.g., eigenvectors and means from vector quantization). Examples abound. In visual processing, it has been shown that prototypical models are capable of capturing intra-class variations such as appearance [26] and lighting [3]. In graphics, prototypes are learned from images and utilized for synthesizing videos for animation [16]. Prototypical representations have also been exploited in signal processing, clustering, dimensionality reduction, dictionary learning compressive sensing, visual tracking and motion analysis , to name a few [5]. In this paper, we assume that samples from one class can be modeled with a small set of prototypes from the same class.

Among the above-mentioned prototype learning algorithms, the method of optimal directions (MOD) [15] and K-SVD algorithms [14] are of great interest as they are able to represent each sample with a sparse combination of dictionary atoms or prototypes. Assuming that we are given a learned dictionary, we can first approximate the basis vectors with sparse representation using the prototypes in this dictionary. As will be explained later in this paper, the original sparse representation problem can then be reduced to a much smaller problem with ℓ_1-norm constraints. By exploiting the linear constraints of these prototypes and basic concepts in linear algebra, we show that the original ℓ_1-norm minimization problem can be reduced from a *large* and *dense* linear system to a *small* and *sparse* one, thereby obtaining significant speed-up. We apply the proposed algorithm to several large data sets and demonstrate that it is able to achieve double-digit gain in speed with much less memory requirement than the state-of-the-art sparse coding methods.

2. Sparse Representation with Prototypes

In numerous problems we are often given a set of labeled samples for each class and the goal is to correctly infer the class of unseen samples by using the knowledge learned from the given samples. Suppose that we collect n_i samples from the i-th of K distinct signal classes, we define a matrix $\Phi_i \in \mathbb{R}^{m \times n_i}$ for class i as columns of samples:

$$\mathbf{\Phi}_i = [\boldsymbol{\phi}_{i,1}, \boldsymbol{\phi}_{i,2}, \cdots, \boldsymbol{\phi}_{i,n_i}], \tag{1}$$

where $\boldsymbol{\phi}_{i,j} \in \mathbb{R}^m$ stands for the j-th sample of the class i. We then concatenate all samples for all K classes into a matrix $\mathbf{\Phi} \in \mathbb{R}^{m \times N}$:

$$\mathbf{\Phi} = [\mathbf{\Phi}_1, \mathbf{\Phi}_2, \cdots, \mathbf{\Phi}_K], \tag{2}$$

where N is the total number of samples from all classes. Given a sufficient number of samples for class i, an observed sample $\mathbf{y}$ can be well approximated by a linear combination of the samples if $\mathbf{y}$ belongs to class i:

$$\mathbf{y} = c_{i,1}\boldsymbol{\phi}_{i,1} + c_{i,2}\boldsymbol{\phi}_{i,2} + \cdots + c_{i,n_i}\boldsymbol{\phi}_{i,n_i}, \tag{3}$$

where the scalar $c_{i,j}$ represents the weighted contribution of the j-th sample in reconstructing the observed sample $\mathbf{y}$. However, we do not know which class the sample $\mathbf{y}$ belongs to in most circumstances. Thus, we can rewrite the linear representation of $\mathbf{y}$ using all samples compactly as:

$$\mathbf{y} = \mathbf{\Phi}\mathbf{x}, \tag{4}$$

where $\mathbf{x} = [0, \cdots, 0, c_{i,1}, \cdots, c_{i,n_i}, 0, \cdots, 0]^\top$ is a sparse coefficient vector. Usually, $\mathbf{\Phi}$ is a fat and dense matrix as illustrated in Figure 1(a).

2.1. Solving Inverse Linear System

With the formulation in (4), each observed sample $\mathbf{y}$ can be represented with the corresponding coefficient vector $\mathbf{x}$ by solving the linear system $\mathbf{y} = \mathbf{\Phi}\mathbf{x}$. If the dimension of the observation data $\mathbf{y}$ is larger than the number of all samples (i.e., $m > N$), then the unique solution can usually be obtained by solving the overdetermined system. However, in most applications, the linear systems are ill-conditioned or underdetermined, resulting in infinitely many solutions to this inverse problem (as shown in Figure 1(a)). Therefore, regularization constraints are of critical importance for obtaining useful solutions. For example, solutions can be obtained by solving the following minimum ℓ_2-norm problem:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_2 \quad \text{subject to} \quad \mathbf{y} = \mathbf{\Phi}\mathbf{x}, \tag{5}$$

and the minimum ℓ_2-norm solution can be obtained by $\hat{\mathbf{x}}_2 = (\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi})^{-1}\mathbf{\Phi}^{\mathrm{T}}\mathbf{y}$. However, the minimum ℓ_2-norm solution $\hat{\mathbf{x}}_2$ is usually dense (i.e., with many nonzero entries), thereby losing the discriminative ability to select the most relevant samples for representing the observed sample $\mathbf{y}$.

Since an observed sample is assumed to belong to one certain class, it can usually be well represented using other samples from that class. Such property has been exploited extensively in the literature, e.g., local linear embedding, image clustering, spectral methods, and face recognition [5]. With a sufficiently large number of samples for each class, the coefficient vector $\mathbf{x}$ is expected to be very sparse, i.e., only a small portion of entries are nonzero. Regularized via sparsity constraints, we seek a representation for an observed sample $\mathbf{y}$:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad \text{subject to} \quad \mathbf{y} = \mathbf{\Phi}\mathbf{x}, \tag{6}$$

where $\|\cdot\|_0 : \mathbb{R}^N \rightarrow \mathbb{R}$ counts the number of nonzero entries. However, solving the ℓ_0-norm minimization of an underdetermined system is both numerically unstable and NP-hard [5].

Recently, theories developed from sparse representation and compressive sensing [8, 9, 13] suggest that if the solution of $\mathbf{x}$ is sparse enough, then the sparsest solution can be recovered via the ℓ_1-norm minimization:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{subject to} \quad \mathbf{y} = \mathbf{\Phi}\mathbf{x}, \tag{7}$$

where the ℓ_1-norm sums up the absolute weights of all entries in $\mathbf{x}$. Note that the equality constraint in (7) can be relaxed to allow small noise, and the sparest solution $\mathbf{x}_0$ can be approximately recovered by finding the minimum ℓ_1-norm vector, $\mathbf{x}$, that best explains the observed sample $\mathbf{y}$:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{subject to} \quad \|\mathbf{y} - \mathbf{\Phi}\mathbf{x}\|_2 \leq \epsilon, \tag{8}$$

where ϵ is the allowed error tolerance. The problems of solving (7) and (8) are convex programs which can be solved by recasting them as linear programs (LP) and second-order cone programs (SOCP) [7, 5], respectively.

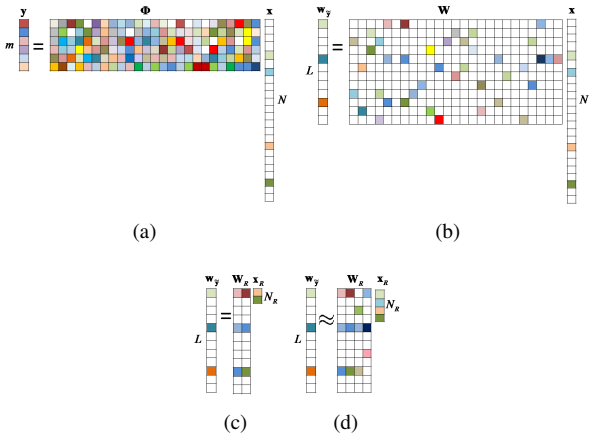


Figure 1. Sparse representation algorithms. (a) The original problem that solves $\mathbf{y} = \mathbf{\Phi}\mathbf{x}$ where $\mathbf{\Phi}$ is a dense fat matrix. (b) The proposed method solves $\mathbf{w}_{\bar{\mathbf{y}}} = \mathbf{W}\mathbf{x}$ where $\mathbf{W}$ is a tall sparse matrix. (c) The proposed method further reduces $\mathbf{W}$ to a tall skinny matrix $\mathbf{W}_R$ and solves $\mathbf{w}_{\bar{\mathbf{y}}} = \mathbf{W}_R\mathbf{x}_R$. (d) The proposed method reduces the matrix $\mathbf{W}$ to a tall skinny matrix $\mathbf{W}_R$ with relaxed constraints and solves $\mathbf{w}_{\bar{\mathbf{y}}} = \mathbf{W}_R\mathbf{x}_R$.

2.2. Feature Extraction by Linear Transformation

Since directly operating on the original space of image observations is computationally expensive due to extremely high data dimensions, numerous feature extraction methods have been proposed to project the original data onto a low dimensional feature space. Thanks to the fact that most feature extraction methods require or can be approximated by linear transformations, the mapping from the image observation space to the feature space can be characterized by a matrix $\mathbf{T} \in \mathbb{R}^{d \times m}$, where $d << m$. For example, $\mathbf{T}$ can be the projection matrix obtained from principal component analysis or simply a downsampling matrix.

Applying $\mathbf{T}$ on both sides of (4), we have

$$\widetilde{\mathbf{y}} = \mathbf{T}\mathbf{y} = \mathbf{T}\mathbf{\Phi}\mathbf{x} = \mathbf{F}\mathbf{x}, \tag{9}$$

where $\widetilde{\mathbf{y}} = \mathbf{T}\mathbf{y}$ is the feature vector of the observed sample $\mathbf{y}$ and $\mathbf{F} = \mathbf{T}\mathbf{\Phi} = [\mathbf{f}_{1,1}, \mathbf{f}_{1,2}, \cdots, \mathbf{f}_{i,n_i}, \cdots, \mathbf{f}_{K,n_K}]$ contains the feature vectors of all samples. As the system of linear equations $\widetilde{\mathbf{y}} = \mathbf{F}\mathbf{x}$ is underdetermined and the solution $\mathbf{x}$ is expected to be sparse, we can recover the solution $\mathbf{x}$ by solving an ℓ_1-norm minimization problem similar to (8):

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{subject to} \quad \|\widetilde{\mathbf{y}} - \mathbf{F}\mathbf{x}\|_2 \leq \epsilon. \tag{10}$$

3. Fast Sparse Approximation with Prototypes

While sparse representations have demonstrated much success in numerous applications [17, 14, 22, 30, 34, 33, 5], solving the convex programs in (7) or (8) remains a challenging problem [11, 13, 5, 32]. Furthermore, a sufficiently large number of samples are often necessary in order to obtain good approximation. The high computational overhead obstructs sparse representations for large-scale real-world problems.

In this section, we present a fast algorithm by exploiting the structure of the basis matrix $\mathbf{F}$ using sparse approximation. The proposed algorithm is motivated by the recent advances in designing overcomplete dictionary for sparse signal representation [2, 5]. Using either pre-defined dictionaries (e.g., overcomplete DCT and wavelets) or adaptive dictionaries learned from data (e.g., K-SVD [2] and MOD [15]), the basis matrix $\mathbf{F}$ consisting of feature vectors of all samples can be well approximated by a linear combination of a small number of dictionary atoms with their corresponding sparse coefficients. With sparse approximation on the basis matrix $\mathbf{F}$, the problem becomes searching for the *sparsest solution* on a *sparse* dictionary. Consequently, the original ℓ_1-norm minimization problem with a large and dense matrix in (10) can be reduced to one with a small and sparse matrix, which can be solved very efficiently compared to the original one. We first briefly describe the most relevant works on dictionary learning for sparse representation, and then present how this can be utilized for fast ℓ_1-norm minimization.

3.1. Learning Overcomplete Dictionary for Sparse Representation

Sparse and redundant modeling of signals has been proven to be very effective for signal reconstruction and classification. Using an overcomplete dictionary $\mathbf{D} \in \mathbb{R}^{d \times L}$ that contains L prototypes or atoms as column vectors, the signal $\mathbf{f} \in \mathbb{R}^d$ can be represented (or well approximated) by a sparse linear combination of these atoms. Specifically, there exists a sparse coefficient vector $\mathbf{w}$ such that the reconstruction of $\mathbf{f}$ can be either exact $\mathbf{f} = \mathbf{D}\mathbf{w}$, or approximated $\mathbf{f} \approx \mathbf{D}\mathbf{w}$. Numerous algorithms have been proposed for the design of dictionaries, including predefined and adaptive ones. Predefined dictionaries include overcomplete DCT, wavelets, curvelts, contourlets, steerable wavelets filters, short-time Fourier transforms, etc [5]. Recently, adaptive dictionary learning algorithms have been shown to achieve superior performance over the predefined dictionaries in several image processing applications, including denoising [14], compression [6], inpainting [23, 24], and super resolution [34]. Among all the existing dictionary learning algorithms [27, 28, 21, 18, 15, 2], the recently proposed K-SVD [2] is one of the most efficient algorithms due to its simplicity and effectiveness.

Given a set of samples $\{\mathbf{f}_i\}_{i=1}^N$, the K-SVD algorithm finds the best dictionary $\mathbf{D}$ to represent the samples as sparse decompositions by minimizing the reconstruction error in ℓ_2-norm:

$$\min_{\mathbf{D},\mathbf{W}} \|\mathbf{F} - \mathbf{D}\mathbf{W}\|_F^2 = \sum_{i=1}^{K} \sum_{j=1}^{n_i} \|\mathbf{f}_{i,j} - \mathbf{D}\mathbf{w}_{i,j}\|_2^2 \quad \text{subject to} \quad \|\mathbf{w}_{i,j}\|_0 \leq S_0, \tag{11}$$

where $\mathbf{w}_{i,j}$ is the sparse representation for j-th samples of class i, and S_0 indicates the maximum allowed nonzero entries in $\mathbf{w}_{i,j}$ (i.e., the coding length). The sparsification process alternates between the sparse coding and the dictionary update stages iteratively to minimize the objective function in (11). The detailed derivations of the K-SVD algorithm can be found in [2].

3.2. Solving Equivalent ℓ_1-norm Minimization Problems with Prototype Constraints

Assume that we have learned the dictionary $\mathbf{D}$ from a set of samples, we can then approximate the matrix $\mathbf{F}$ of (9) with $\mathbf{D}$ and its sparse representation $\mathbf{W}$ from (11). For a new observation $\widetilde{\mathbf{y}}$, we can find its atom decomposition (i.e., the sparsest representation) over the learned dictionary $\mathbf{D}$. Then, the system of linear equations in the feature space (9) can be rewritten as

$$\widetilde{\mathbf{y}} \approx \mathbf{D}\mathbf{w}_{\widetilde{\mathbf{y}}} \approx \mathbf{D}\mathbf{W}\mathbf{x}. \tag{12}$$

If the learned dictionary $\mathbf{D}$ is capable of approximating the signals $\widetilde{\mathbf{y}}$ well, i.e., $\|\widetilde{\mathbf{y}} - \mathbf{D}\mathbf{w}_{\widetilde{\mathbf{y}}}\|_2 \leq \epsilon$ for a small constant ϵ,

we can represent the signal $\widetilde{\mathbf{y}}$ as $\widetilde{\mathbf{y}} = \mathbf{D}\mathbf{w}_{\widetilde{\mathbf{y}}} + \mathbf{e}_{\widetilde{\mathbf{y}}}$, where $\mathbf{e}_{\widetilde{\mathbf{y}}}$ is the residual with $\|\mathbf{e}_{\widetilde{\mathbf{y}}}\|_2 \leq \epsilon$. Similarly, the matrix $\mathbf{F}$ can be expressed as $\mathbf{F} = \mathbf{D}\mathbf{W} + \mathbf{e}_{\mathbf{F}}$, where $\mathbf{e}_{\mathbf{F}} \in \mathbb{R}^{d \times N}$ is the residual and $\|\mathbf{e}_{\mathbf{F}}\|_F \leq \sqrt{N}\epsilon$. By introducing the residual signals, we can rewrite (12) as

$$\mathbf{D}\mathbf{w}_{\widetilde{\mathbf{y}}} + \mathbf{e}_{\widetilde{\mathbf{y}}} = \mathbf{D}\mathbf{W}\mathbf{x} + \mathbf{e}_{\mathbf{F}}\mathbf{x} \Longrightarrow \mathbf{D}(\mathbf{w}_{\widetilde{\mathbf{y}}} - \mathbf{W}\mathbf{x}) = \mathbf{e}_{\mathbf{F}}\mathbf{x} - \mathbf{e}_{\widetilde{\mathbf{y}}}. \quad (13)$$

Recall that the solution $\mathbf{x}$ is assumed to be sparse, say s-sparse (i.e., only s entries are non-zeros), we have

$$\|\mathbf{D}(\mathbf{w}_{\widetilde{\mathbf{y}}} - \mathbf{W}\mathbf{x})\|_2 \leq (s+1)\epsilon. \quad (14)$$

Let $\mathbf{z} = \mathbf{w}_{\widetilde{\mathbf{y}}} - \mathbf{W}\mathbf{x}$, which is also a sparse vector[1], we have $\|\mathbf{D}\mathbf{z}\|_2 \leq (s+1)\epsilon$. Using the *restricted isometry property* (RIP) [9], we can determine whether the sparse coding with a dictionary can be *stably* obtained. Recall a matrix $\mathbf{D}$ satisfies the RIP of order $s_{\mathbf{z}}$ with constant $\rho = \rho_{s_{\mathbf{z}}} < 1$ if

$$\|\mathbf{z}\|_0 \leq s_{\mathbf{z}} \Longrightarrow (1-\rho)\|\mathbf{z}\|_2^2 \leq \|\mathbf{D}\mathbf{z}\|_2^2 \leq (1+\rho)\|\mathbf{z}\|_2^2. \quad (15)$$

Suppose that $\mathbf{z}$ is $s_{\mathbf{z}}$-sparse and the dictionary satisfies the RIP of order $s_{\mathbf{z}}$, we can derive an upper-bound for $\mathbf{z}$ using (14) and (15):

$$(1-\rho)\|\mathbf{z}\|_2^2 \leq \|\mathbf{D}\mathbf{z}\|_2^2 \leq (s+1)^2\epsilon^2, \quad (16)$$

and thus

$$\|\mathbf{z}\|_2 = \|\mathbf{w}_{\widetilde{\mathbf{y}}} - \mathbf{W}\mathbf{x}\|_2 \leq \frac{(s+1)\epsilon}{\sqrt{(1-\rho)}} = \widetilde{\epsilon}. \quad (17)$$

The exact value of RIP constant is unknown (as computing the value is an NP-hard problem). However, suppose that the RIP holds, $\mathbf{D}$ approximately preserves the Euclidean length of $s_{\mathbf{z}}$-sparse signal. Thus we know that $\|\mathbf{z}\|_2$ is upper-bounded by a certain constant value, thereby ensuring that the sparse solution $\mathbf{x}$ can be approximately recovered by solving a much smaller problem. That is, the solution $\mathbf{x}$ can now be obtained by solving the following equivalent ℓ_1-minimization problem:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{subject to} \quad \|\mathbf{w}_{\widetilde{\mathbf{y}}} - \mathbf{W}\mathbf{x}\|_2 \leq \widetilde{\epsilon}. \quad (18)$$

If the dictionary $\mathbf{D}$ is assumed to provide exact reconstruction for signals (i.e., allowing enough non-zero entries in the coefficient vectors), then $\widetilde{\epsilon} = 0$ and the problem in (18) is reduced to

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{subject to} \quad \mathbf{w}_{\widetilde{\mathbf{y}}} = \mathbf{W}\mathbf{x}, \quad (19)$$

as illustrated in Figure 1(b).

[1] The vector $\mathbf{z}$ is based on two closely related sparse coefficients. In the case of relaxed constraint, generally the vector $\mathbf{z}$ would be sparse as long as the solution $\mathbf{x}$ is sufficiently sparse, say with s non-zero entries. Denote the sparsity of the coefficient vector $\mathbf{w}_{\widetilde{\mathbf{y}}}$ and the columns of matrix $\mathbf{W}$ as $s_{\mathbf{w}}$. Then, we can derive an upper-bound for the sparsity of the vector $\mathbf{z}$: $s_{\mathbf{z}} <= s_{\mathbf{w}} + s(s_{\mathbf{w}} - 1)$, which is the case that for each column in $\mathbf{W}$ chosen, there is only one support match to $\mathbf{w}_{\widetilde{\mathbf{y}}}$ (worst case).

We now present how the reduced ℓ_1-minimization problem can recover the same sparse solution with significant gain in speed than existing algorithms. First consider the exact reconstruction case in (19), where $\mathbf{W}$ is now of dimension $L \times N$ instead of $d \times N$. At first glance, we have a larger linear programming problem to solve. However, since $\mathbf{W}$ contains only sparse column vectors, the equation in (19) can be significantly reduced by identifying only the relevant columns that have the same supports as $\mathbf{w}_{\widetilde{\mathbf{y}}}$, i.e., the sparse representation of observed sample $\widetilde{\mathbf{y}}$. The identification of such columns (i.e., prototypes of each sample class) and matrix reduction process are as follows. Given $\mathbf{w}_{\widetilde{\mathbf{y}}}$, we first locate nonzero entries in $\mathbf{w}_{\widetilde{\mathbf{y}}}$, denoted as $\alpha = \{i | \mathbf{w}_{\widetilde{\mathbf{y}}}(i) \neq 0\}$ and $|\alpha| = N_R$ (as shown in Figure 1(b)). We then sweep all the columns in $\mathbf{W}$ to check whether the nonzero entries in each column vector matches the support of $\mathbf{w}_{\widetilde{\mathbf{y}}}$. For those columns with non-identical supports, there is by no chance that these columns will be used in representing the $\mathbf{w}_{\widetilde{\mathbf{y}}}$ (a straightforward result of the column space from linear algebra). We can simply discard these columns and set the corresponding coefficient values in $\mathbf{x}$ to zero, and have a reduced matrix $\mathbf{W}_R$ which is usually much smaller than $\mathbf{W}$. For example, suppose $\mathbf{w}_{\widetilde{\mathbf{y}}}$ has only three nonzero entries at its 1st, 4th, and 9th elements. We will sweep over all the columns of $\mathbf{W}$ and retain only those vectors with same supports (i.e., whose 1st, 4th, and 9th elements are nonzero) as shown in Figure 1(b). After identifying columns that may be used for representing $\mathbf{w}_{\widetilde{\mathbf{y}}}$, we can have the same exact solution by solving the reduced ℓ_1-norm minimization:

$$\min_{\mathbf{x}_R} \|\mathbf{x}_R\|_1 \quad \text{subject to} \quad \mathbf{w}_{\widetilde{\mathbf{y}}} = \mathbf{W}_R\mathbf{x}_R, \quad (20)$$

where $\mathbf{W}_R \in \mathbb{R}^{L \times N_R}$ contains only relevant columns (with the same support of $\mathbf{w}_{\widetilde{\mathbf{y}}}$) and $\mathbf{x}_R \in \mathbb{R}^{N_R}$, as shown in Figure 1(c). As N_R is usually much smaller than N, the resulting matrix $\mathbf{W}_R$ is a skinny matrix.

3.3. Relaxed Prototype Constraints

As for the second case in (18), we use the same identification and reduction process to discard irrelevant column vectors. However, since we allow small error $\widetilde{\epsilon}$ in reconstructing $\mathbf{w}_{\widetilde{\mathbf{y}}}$, columns with partial overlapping supports as $\mathbf{w}_{\widetilde{\mathbf{y}}}$ can also be used in representing $\mathbf{w}_{\widetilde{\mathbf{y}}}$. In other words, we relax the prototype constraints without using exactly the same supports. Our motivation is that if the number of overlapped entries of a certain column vector with $\mathbf{w}_{\widetilde{\mathbf{y}}}$ is small, the likelihood of this column vector being used for representing $\mathbf{w}_{\widetilde{\mathbf{y}}}$ is low as the reconstruction cost when selecting this column is higher. Therefore, we propose a set of approximation criteria for reducing the problem in (18). We denote these approximation criteria as $\{R_j\}_{j=1}^{J}$, where j indicates the number of minimal allowed overlapped entries with the supports of $\mathbf{w}_{\widetilde{\mathbf{y}}}$ and columns in $\mathbf{W}$. For example, if $j = 2$, then $\mathbf{W}_{R_j}$ contains columns with supports that have

at least 2 overlapped entries with $\mathbf{w}_{\tilde{\mathbf{y}}}$. , as illustrated in Figure 1(d). As j increases, the number of columns in $\mathbf{W}_{R_j}$ decreases, resulting in a faster minimization process. However, it may introduce errors when j is large. It is actually a trade-off between speed performance and accuracy. Nevertheless, one can still have the solution as (19) by simply discarding those columns with no overlapped entries with $\mathbf{w}_{\tilde{\mathbf{y}}}$.

3.4. Time and Space Complexity

There are three main steps in solving the reduced problem in (20). First, the matrix $\mathbf{w}_{\tilde{\mathbf{y}}}$ can be computed efficiently using OMP with $O(dLN_R)$ flops [5, 31]. Second, $\mathbf{W}$ can be computed from (11) using the K-SVD algorithm which takes $O(LN_R^2 + dL)$ flops [5, 31]. It then takes one sweep over the columns of $\mathbf{w}_{\tilde{\mathbf{y}}}$ in order to obtain $\mathbf{W}_R$, and the time complexity is $O(LN)$. The state-of-the-art algorithm for ℓ_1-norm minimization, $\mathbf{Ax} = \mathbf{b}$, recasts the original problem as a SOCP which is then solved by the log-barrier algorithm [7, 12, 32]. At the core of the log-barrier algorithm, it solves a linear system with quadratic constraints formed by $\mathbf{A}^\top\mathbf{A}$ using the conjugate gradient method [7, 12, 32]. That is, the computational costs for solving (4) and (20) hinge on $\mathbf{\Phi}^\top\mathbf{\Phi}$ and ${\mathbf{W}_R}^\top\mathbf{W}_R$ [4]. Thus, the time complexity of the matrix multiplication for a dense matrix $\mathbf{\Phi}$ and $\mathbf{W}_R$ is $O(m^2N^2)$ flops and $O(L^2N_R^2)$ flops, respectively. The time complexity ratio between the original and proposed method is mainly determined by the quadratic terms, i.e., $O(\frac{m^2N^2}{L^2N_R^2+dLN_R+LN_R^2+dL+LN}) \approx O(\frac{N^2}{N_R^2})$, as N_R is much smaller than N, and L and m are usually of the same scale. As a result, the proposed algorithm achieves significant quadratic speed-up.

In the intermediate step, the space complexity of the proposed algorithm for storing the learned dictionary and basis matrix $\mathbf{W}$ of (12) is $O(LK + KN)$. However, the space complexity for the reduced matrix $\mathbf{w}_{\tilde{\mathbf{y}}}$ is $O(N_R)$. As the quadratic constraints of $\mathbf{\Phi}^\top\mathbf{\Phi}$ and ${\mathbf{W}_R}^\top\mathbf{W}_R$ are computed in solving ℓ_1-minimization, the original formulation again has much higher space complexity than the proposed one, with the ratio of $O(\frac{N^2}{LK+KN+N_R^2}) \approx O(\frac{N^2}{N_R^2})$.

4. Experimental Results

In this section, we present experimental results on both synthetic and real data sets to demonstrate the efficiency and effectiveness of the proposed algorithm. All the experiments were carried out using MATLAB implementations to solve the original and proposed relaxed ℓ_1-norm minimization problems described in Section 2.2 as well as Section 3.3 on a 1.8 GHz machine with 2 GB RAM. The MATLAB code and processed data is available at `faculty.ucmerced.edu/mhyang/fsr.html`.

4.1. Synthetic Data

We validate the effectiveness of sparse representations and the efficiency of the proposed approximation for signal classification in the presence of Gaussian white noise. We first build a dictionary with all elements drawn from a Gaussian random variable with zero mean and unit variance. The columns of the dictionary are then normalized to unit norm. Since samples from one specific class are assumed to lie in certain subspace that can be modeled by prototypes, we generate samples by first selecting five columns in the dictionary and then obtain each sample by a linear combination of the selected columns. In the first experiment, we generate 50 samples for each of the 10 classes, where the dimension of each sample is 25. The test samples, assumed to lie in one unknown subspace, are generated by randomly selecting one signal class and combining three training samples from the selected class with random coefficients. Gaussian noise of different levels are added to the test samples for experiments. We generate 100 test samples to evaluate the recognition capability of the classification based on sparse representation. That is, for each test sample, we need to solve (10) with matrix $\mathbf{F} \in \mathbb{R}^{25\times500}$ for recognition (using their sparse coefficients to find the class with minimum reconstruction error). In this experiment, we use the K-SVD algorithm to learn the underlying dictionary $\mathbf{D} \in \mathbb{R}^{25\times100}$ from $\mathbf{F}$ although other algorithms such as MOD can be substituted. In the sparse coding stage, OMP with sparsity factor 5 (i.e., the maximum allowed nonzero coefficients) is used. After 10 iterations of the dictionary learning process, we compute the sparse representations of samples in $\mathbf{F}$ and $\tilde{\mathbf{y}}$. With these sparse representations, we can obtain the approximated solution of $\mathbf{x}$ by solving (18).

In Figure 2, we report the recognition rates of methods with sparse representation from solving (10) and (18) where the label is determined based on minimum reconstruction error. It shows that these methods are able to recognize correct classes even under substantial noise corruption (up to 20%). Furthermore, the classifier with the proposed algorithm (using (18)) achieves higher recognition rate than the one with original one (using (10)). This is because that when samples in $\mathbf{F}$ are coded with sparse representation using dictionary $\mathbf{D}$, noise in the signals are also removed as a by-product of ℓ_1-norm minimization.

In the second synthetic experiment, we present the runtime performance of the proposed method and the state-of-art ℓ_1-norm optimization solver with SOCP techniques [7]. We increase both the number of signal class and the feature dimension of samples to 50. Following similar procedure as described in the first experiment, we generate three sets of training samples (500, 1000, 1500). Like the previous experiment, 100 test samples are obtained in the same way as previous setting with noise of zero mean and σ^2 is set to 0.1. Table 1 shows the mean value of the recognition rates

Table 1. Comparison on recognition speed and accuracy using synthetic data sets.

Sample size	500		1000		1500	
Method	Acc (%)	Time (s)	Acc (%)	Time (s)	Acc (%)	Time (s)
Original	96.0	0.7575	96.6	4.4548	97.0	12.3191
Proposed	93.0	0.0061	94.6	0.0114	96.0	0.0217
Speed-up	124.2		390.8		567.7	

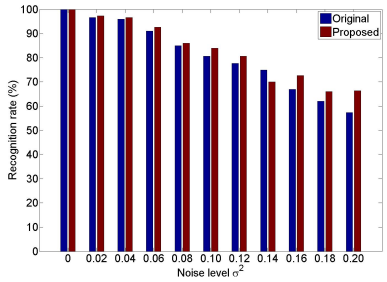


Figure 2. Recognition rates of the original and the proposed algorithm in signal classification task with the presence of noise.

and the average computation time per test sample needed in solving these two equations. Although the classifier with sparse representation obtained by the original l_1-norm minimization has slightly higher recognition rate in this setting, however, the execution time increases rapidly as the number of the samples increases. On the other hand, the classifier with the proposed algorithm achieves comparable results in accuracy but with significant speed-up. For the set of 1500 samples, our method solves the l_1-norm minimization problem 567.7 times faster than the l_1-magic solver (using log barrier algorithm for SOCP) [7]. The reason that the proposed method achieves slightly worse results in accuracy may result from the imperfect approximation of the sparse approximation using the K-SVD algorithm. We note that this is a trade-off between accuracy and speed that can be adjusted with parameters in the K-SVD algorithm (e.g., the number of coefficients in OMP). It is also worth noticing that l_1-magic solver can handle up to 4500 data points (of 50 dimensions) whereas our algorithm is able to handle more than 10000 data points.

4.2. Face Recognition

We use the extended Yale database B which consists of 2414 frontal face images of 38 subjects for experiments [20]. These images were acquired under various lighting conditions and normalized to canonical scale (as shown in Figure 3). For each subject, we randomly select half of the images as training samples and use the rest for tests.

Figure 3. Sample images from the extended Yale database B.

Since the original data dimension is very high ($192 \times 168 = 32256$), we use two feature extraction methods (downsampling and PCA) to reduce their dimensionality (i.e. difference choice of $\mathbf{T}$ in (9)). In the appearance-based face recognition setup each image is downsampled to 12×11 pixels (i.e., $d = 132$). For PCA, we compute the eigenfaces using the training images and retain the coefficients of the largest 132 eigenvectors. A dictionary of size 132×264 (i.e., redundancy factor of 2) is trained using the K-SVD algorithm with 10 iterations. For each sample, at most 10 coefficients are used for sparse coding with OMP.

As shown in Table 2, the proposed algorithm achieves comparable recognition rates but with significant speed-up. Note that the required time to classify one image using the original l_1-norm minimization is more than 20 seconds, which makes it infeasible for real-world applications. By increasing the redundancy of the dictionary or reducing the maximum allowed number of coefficients for in representing an image (i.e., making the representation even more sparser), we can achieve more speed-ups at the expense of slight performance degradation.

Table 2. Recognition accuracy and speed using the Extended Yale database B.

Feature	Downsampled image		PCA	
Method	Acc (%)	Time (s)	Acc (%)	Time (s)
Original	93.78	20.08	95.01	13.17
Proposed	91.62	0.51	92.28	0.32
Speed-up	39.4		41.2	

4.3. Single Image Super-Resolution

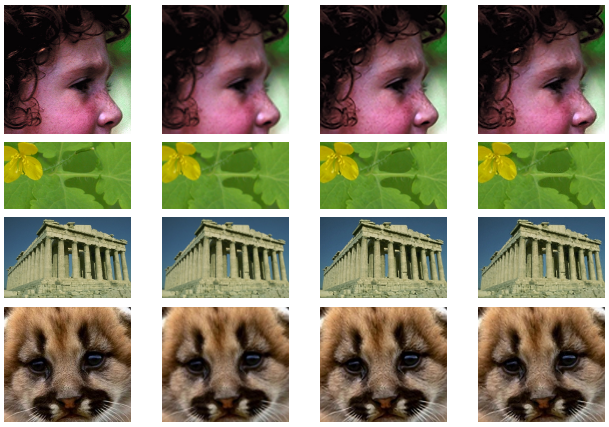
We apply the proposed algorithm to image super resolution using sparse representation [34], which assumes sparsity prior for patches from a high-resolution image. Two dictionaries, one for the low-resolution image and the other for the high-resolution one, are trained using patches randomly selected from an image collection. The reconstructed high resolution image can then be obtained by solving ℓ_1-norm penalized sparse coding. Here the original ℓ_1-norm penalized sparse coding adopted is based on the efficient sparse coding algorithm in [19]. The dictionary size is set 6 times the feature dimensions in the low-resolution dictionary. For each sample, 3 coefficients are used for spare approximation by OMP.

In Table 3, we report the root-mean-square error (RMSE) values in pixel intensity and the execution time of sparse

coding for all patches on four test images used in [34]: Girl, Flower, Panthenon, and Racoon. The proposed algorithm achieves double-digit speedups with slight performance degradation (in terms of RMSE). However, the RMSE measure is not the best metric for super resolution as it does not directly reflect the visual quality and oftentimes we do not have ground truth high-resolution images. Figure 4 shows visual quality of the proposed algorithm is much better than the results using bicubic interpolation, and very similar to the results of [34].

Table 3. Execution time and RMSE for sparse coding on four test images (scale factor = 3)

Image	Original [19]		Proposed		
Method	RMSE	Time (s)	RMSE	Time (s)	Speedup
Girl	5.6684	17.2333	6.2837	1.5564	11.07
Flower	3.3649	14.9173	3.8710	1.3230	11.27
Panthenon	12.247	35.1163	13.469	3.1485	11.15
Racoon	9.3584	27.9819	10.148	2.3284	12.02



(a) (b) (c) (d)

Figure 4. Image super-resolution visual results. (a) Ground-truth high resolution image. (b) Bicubic interpolation. (c) Super-resolution via sparse representation [34]. (d) Proposed method.

4.4. Human Pose Estimation

We apply the sparse representation to the problem of estimating human pose from single images. Here we pose this problem as a regression that maps image observations to three-dimensional joint angles. In training set, we are given a number of silhouette images and their corresponding pose parameters. The task is to infer the three-dimensional human pose of an unseen test image.

We validate the applicability of the sparse representation to this regression problem and demonstrate the improvement in speed with the proposed algorithm. We use the INRIA data set [1] in which 1927 silhouette images are used for training and 418 images for tests (some samples are shown in Figure 5). The image descriptors we use are the 100-dimensional feature vectors (i.e., histogram of shape context descriptors) as computed in [1].

Figure 5. Sample images from the INRIA data set [1].

For each test image, we find the sparse representation by solving the l_1-norm minimization problem. That is, the resulting linear combination of training images best represent the test image with minimum reconstruction error (in ℓ_2-norm). The estimated pose of the test image is then computed by the same linear combination of the associated pose parameters in the training set.

We report in Table 4 the execution time of solving the l_1-norm minimization problem with different numbers of prototypes. We note that the mean estimation error of 3D joint angles decreases as the number of prototypes is increased. Overall, the speed can be improved significantly with only slight performance degradation.

4.5. Multi-View Object Recognition

We compare the proposed method against the original algorithm using the Columbia Object Image Library (COIL-100) data set [26]. The COIL-100 data set has been widely used in object recognition literature, and it consists of color images of 100 distinct objects; 72 images of each object placed on a turntable were captured at pose interval of 5 degrees. Typically, a few number of images from different views with constant interval are selected as training samples and the others are used for tests. For example, if 18 views are selected from each object class for experiments, there will be 1800 training and 5400 test images.

The 128×128 images are first converted to gray images and then downsampled to 32×32 pixels. Two kinds of simple features (downsampled image and PCA) are used for evaluations. Meanwhile, two different feature dimensions are used for experiments. The dictionary size is set 4 times the feature dimensions (i.e., $4d$). For each sample, 3 coefficients are used for spare representation by OMP. The recognition rate and the average execution time for processing one sample are summarized in Table 5. Overall, the proposed algorithm achieves about 10000 times faster than the original method with comparable accuracy (note the execution time are recorded in different scales).

5. Conclusion

In this paper, we have presented a fast sparse representation algorithm that exploits the prototype constraints inhere in signals. We show that sparse representation with ℓ_1-norm minimization can be reduced to a smaller linear system and thus significant gains in speed can be obtained. In addition, the proposed algorithm requires much less memory than the state-of-the-art ℓ_1-minimization solver. Experimental

Table 4. Comparison on pose estimation accuracy and speed under different number of prototypes using the INRIA data set.

Number of coefficients	3	6	9	12	15	Original l_1-norm minimization
Mean error (in degrees)	9.1348	7.9970	7.4406	7.2965	7.1872	6.6513
Execution time (in seconds)	0.0082	0.0734	0.3663	1.1020	2.3336	24.69
Speed-up	3011.0	336.4	67.4	22.4	10.6	

Table 5. Comparison on recognition speed and accuracy on the COIL-100 (note some values are listed in different scales).

Number of view	8		16		8			16		
	Recognition accuracy				Execution time					
Feature used	Orig. (%)	Ours (%)	Orig. (%)	Ours (%)	Orig. (s)	Ours (ms)	Speed-up	Orig. (s)	Ours (ms)	Speed-up
Downsampling(16×16)	82.43	80.93	90.01	87.43	6.85	3.2	2140.6	52.73	3.9	13520.5
Downsampling(10×10)	75.08	74.28	84.75	84.00	4.13	3.9	1059.0	48.02	5.2	9234.6
PCA: 256	84.56	82.08	91.03	89.22	3.71	3.3	1124.2	29.58	3.8	7784.2
PCA: 100	81.23	79.23	90.58	87.72	2.54	3.6	705.6	21.00	5.6	3750.0

results on several image and vision problems demonstrate that our algorithm is able to achieve double-digit gain in speed with much less memory requirement and comparable accuracy.

Our future work will focus on extending the proposed fast algorithm for learning discriminative sparse representations for classification problems. We are also interested in analyzing the interplay between RIP assumption and the effectiveness of the proposed method. As the proposed algorithm is able to handle large-scale data collections, we will explore other real-world applications such as image search and visual tracking.

Acknowledgments

The authors would like to thank anonymous reviewers for their comments. This work is partially supported by a UC Merced faculty start-up fund and a gift from Google.

References

[1] A. Agarwal and B. Triggs. Recovering 3d human pose from monocular images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(1):44–58, 2006. 7

[2] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311, 2006. 1, 3

[3] P. N. Belhumeur and D. J. Kriegman. What is the set of images of an object under all possible illumination conditions. *International Journal of Computer Vision*, 28(3):1–16, 1998. 1

[4] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004. 5

[5] A. Bruckstein, D. Donoho, and M. Elad. From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM Review*, 51(1):34–81, 2009. 1, 2, 3, 5

[6] O. Bryt and M. Elad. Compression of facial images using the K-SVD algorithm. *Journal of Visual Communication and Image Representation*, 19(4):270–282, 2008. 3

[7] E. Candes and J. Romberg. *ℓ_1-magic; Recovery of sparse signals via convex programming*. California Institute of Technology, 2005. `http://www.acm.caltech.edu/l1magic/`. 1, 2, 5, 6

[8] E. Candès, J. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics*, 59:1207–1223, 2006. 1, 2

[9] E. Candes and T. Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE Transactions on Information Theory*, 52(12):5406–5425, 2006. 1, 2, 4

[10] S. Chen, S. Billings, and W. Luo. Orthogonal least squares methods and their application to non-linear system identification. *International Journal of Control*, 50(5):1873–1896, 1989. 1

[11] S. Chen, D. Donoho, and M. Saunders. Automatic decomposition by basis pursuit. *SIAM Journal of Scientific Computation*, 20(1):33–61, 1998. 1, 3

[12] D. Donoho and Y. Tsaig. Fast solution of ℓ_1-norm minimization problems when the solution may be sparse. *IEEE Transactions on Information Theory*, 54(11):4789–4812, 2008. 1, 5

[13] D. L. Donoho. For most large underdetermined systems of linear equations the minimal ℓ_l-norm solution is also the sparsest solution. *Communications on Pure and Applied Mathematics*, 59(6):797–829, 2006. 1, 2, 3

[14] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 15(12):3736–3745, 2006. 1, 3

[15] K. Engan, S. O. Aase, and J. H. Husoy. Method of optimal directions for frame design. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 2443–2446, 1999. 1, 3

[16] T. Ezzat, G. Geiger, and T. Poggio. Trainable videorealistic speech animation. In *Proceedings of ACM SIGGRAPH*, pages 388–398, 2002. 1

[17] K. Huang and S. Aviyente. Sparse representation for signal classification. In *NIPS*, volume 19, pages 609–616, 2006. 3

[18] K. Kreutz-Delgado, J. F. Murray, B. D. Rao, K. Engan, T. W. Lee, and T. J. Sejnowski. Dictionary learning algorithms for sparse representation. *Neural Computation*, 15(2):349–396, 2003. 3

[19] H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. pages 801–808, 2006. 6, 7

[20] K.-C. Lee, J. Ho, and D. J. Kriegman. Acquiring linear subspaces for face recognition under variable lighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(5):684–698, 2005. 6

[21] M. S. Lewicki and T. J. Sejnowski. Learning overcomplete representations. *Neural Computation*, 12(2):337–365, 2000. 1, 3

[22] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Discriminative learned dictionaries for local image analysis. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2008. 3

[23] J. Mairal, M. Elad, and G. Sapiro. Sparse representation for color image restoration. *IEEE Transactions on Image Processing*, 17(1):53, 2008. 3

[24] J. Mairal, G. Sapiro, and M. Elad. Learning multiscale sparse representations for image and video restoration. *SIAM Multiscale Modeling and Simulation*, 7(1):214–241, 2008. 3

[25] S. G. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, 1993. 1

[26] H. Murase and S. Nayar. Visual learning and recognition of 3-D objects from appearance. *International Journal of Computer Vision*, 14(1):5–24, 1995. 1, 7

[27] B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996. 1, 3

[28] B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*, 37(23):3311–3325, 1997. 3

[29] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Conference Record of The Twenty-Seventh Asilomar Conference on Signals, Systems, and Computers*, pages 40–44, 1993. 1

[30] D. Pham and S. Venkatesh. Joint learning and dictionary construction for pattern recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2008. 3

[31] R. Rubinstein, M. Zibulevsky, and M. Elad. Efficient implementation of the k-svd algorithm using batch orthogonal matching pursuit. Technical Report CS-2008-08, Technion, 2008. 5

[32] J. A. Tropp and S. J. Wright. Computational methods for sparse solution of linear inverse problems. Technical Report 2009-01, California Institute of Technology, 2009. 1, 3, 5

[33] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):210–227, 2009. 3

[34] J. Yang, J. Wright, T. Huang, and Y. Ma. Image super-resolution as sparse representation of raw image patches. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2008. 3, 6, 7