Cross-Scale Predictive Dictionaries

Vishwanath Saragadam, *Student Member, IEEE*, Xin Li, *Fellow, IEEE*,
and Aswin C. Sankaranarayanan, *Senior Member, IEEE*

Abstract—Sparse representations using data dictionaries provide an efficient model particularly for signals that do not enjoy alternate analytic sparsifying transformations. However, solving inverse problems with sparsifying dictionaries can be computationally expensive, especially when the dictionary under consideration has a large number of atoms. In this paper, we incorporate additional structure on to dictionary-based sparse representations for visual signals to enable speedups when solving sparse approximation problems. The specific structure that we endow onto sparse models is that of a multi-scale modeling where the sparse representation at each scale is constrained by the sparse representation at coarser scales. We show that this cross-scale predictive model delivers significant speedups, often in the range of 10-60×, with little loss in accuracy for linear inverse problems associated with images, videos, and light fields.

Index Terms—Computational and artificial intelligence, Image processing, Image representation, Sparse representations, Orthogonal Matching Pursuit, Overcomplete dictionary, Multiscale modeling

I. INTRODUCTION

Images are strongly correlated across scales, a fact that is often modeled and exploited to enhance image processing algorithms [1], [2]. An important example of this idea is the wavelet tree model which provides a sparse as well as a *predictive* model for the occurrence of non-zero wavelet coefficients across spatial scales [3]. The wavelet tree model arranges the wavelet coefficients of an image onto a tree whose nodes correspond to the coefficients and each level corresponds to coefficients associated with a particular scale. Under such an organization, the dominant non-zero coefficients form a connected rooted sub-tree [4], i.e., children of a node with small wavelet coefficients are expected to take small values as well. This property has found widespread applicability in tasks like compression [5], sensing [6], [7], and processing [4]. While the wavelet tree model provides excellent approximation capabilities for images, similar models with cross-scale predictive property are largely unknown for other visual signals including videos, hyperspectral images, and light fields.

Overcomplete dictionaries provide an alternate approach for enabling sparse representations [8]. Given a large amount of data, we can *learn* a dictionary such that the training dataset can be expressed as a sparse linear combination of the elements/atoms of the dictionary. The reliance on learning, as opposed to analytic constructions as in the case of wavelets, provides immense flexibility towards obtaining a dictionary that is tuned to the specifics of a particular signal class. Overcomplete dictionaries have found wide applicability for

Fig. 1: Left to right: Bayer image, image reconstructed using OMP, and image reconstructed using the proposed method. While OMP takes 16 minutes, the proposed method takes only 1.5 minutes with little loss in reconstruction quality.

sensing and processing of images [9], videos [10], light fields [11], and other visual signals [12]. However, not much attention has been paid to the incorporation of predictive models to enable speed ups by exploiting correlations across spatial, temporal and angular scales.

In this paper, we propose incorporation of cross-scale predictive modeling in sparsifying dictionaries, thus combining class-specific adaptation with speed ups offered by predictive models. Our specific contributions are as follows.

- **Model.** We propose a novel signal model that uses multi-scale sparsifying dictionaries to provide cross-scale prediction for a wide array of visual signals. Specifically, given the set of sparsifying dictionaries — one for each scale — the non-zero support patterns of a signal and its downsampled counterparts are constrained to only exhibit specific pre-determined patterns.
- **Computational speedups.** The proposed signal model, with its constrained support pattern across scales, naturally enables cross-scale prediction that can be used to speedup the runtime of algorithms like orthogonal matching pursuit (OMP) [13]. Figure 1 shows speed-ups obtained for demosaicing of images; here, we obtain a 10× speed up with little loss in accuracy over a similar-sized dictionary.
- **Learning.** Given large collections of training data, we propose a simple training method, which is modified from the classical K-SVD algorithm [9], to obtain dictionaries that are consistent with our proposed model.
- **Validation.** We verify empirically that the model works through simulation on an array of visual signals including images, videos, and light field images.

A shorter version of this paper appeared at the IEEE International Conference on Image Processing [14]. This journal paper extends the results to a larger class of signals including light fields as well as shows results on real data captured from compressive imaging hardware.

V. Saragaram and A. C. Sankaranarayanan are with the ECE Department at the Carnegie Mellon University.

X. Li is with the ECE Department at the Duke Kunshan University.

II. Prior work

A. Notation

We denote vectors in bold font and scalars/matrices in capital letters. A vector is said to be K-sparse if it has at most K non-zero entires. The support of a sparse vector $\mathbf{s}$, denoted as $\Omega_{\mathbf{s}}$, is the set of the indices of its non-zero entries. The ℓ_0-norm of a sparse vector is the number of non-zero entries or equivalently the cardinality of its support. Finally, given a dictionary $D \in \mathbb{R}^{N \times T}$ and a support set Ω, $D_{|\Omega}$ refers to the matrix of size $N \times |\Omega|$ formed by selecting columns of D corresponding to the elements of Ω; similarly, given a vector $\mathbf{s}$, $\mathbf{s}_{|\Omega}$ refers to an $|\Omega|$-dimensional vector formed by selecting entries in $\mathbf{s}$ corresponding to Ω.

B. Sparse approximation

Sparse approximation problems arise in a wide range of settings [15]. The broad problem definition is as follows: given a vector $\mathbf{x} \in \mathbb{R}^N$, a matrix $D \in \mathbb{R}^{N \times T}$, we solve

$$\text{(P0)} \qquad \min_{\mathbf{s} \in \mathbb{R}^T} \|\mathbf{x} - D\mathbf{s}\|_2 \quad \text{s.t.} \quad \|\mathbf{s}\|_0 \leq K.$$

While the problem itself is NP-hard [16], there are many greedy and relaxed approaches to solving (P0). Of particular interest to this paper is OMP [13], a greedy approach to solving (P0). OMP recovers the support of the sparse vector $\mathbf{s}$, one element at a time, by finding the column of the dictionary that is most correlated with the current residue. In each iteration of the algorithm, there are three steps: first, the index of the atom that is closest in angle to the current residue is added to the support; second, solving a least square problem with the updated support to obtain the current estimate; and third, updating the residue by removing the contribution of the current estimate. Outline of the procedure is given in Algorithm 1.

Algorithm 1 Orthogonal matching pursuit

Require: $\mathbf{x}$, D, K
  $r \leftarrow \mathbf{x}$
  $\Omega \leftarrow \phi$
  $\alpha \leftarrow \phi$
  for $n = 1$ to K **do**
    $k \leftarrow \arg\max_i |\langle \mathbf{d}_i, r \rangle|$ ▷ (Proxy)
    $\Omega \leftarrow \Omega \bigcup k$ ▷ (Support merge)
    $\alpha \leftarrow \arg\min_\beta \|\mathbf{x} - D_{|\Omega}\beta\|^2$ ▷ (Projection)
    $r \leftarrow \mathbf{x} - D_{|\Omega}\alpha$ ▷ (Residue)
  end for
  return Ω, α

The proxy step and the projection step are the two computationally intensive steps in OMP. The time complexity of the proxy step is $\mathcal{O}(NTK)$, while that of the projection step is $\mathcal{O}(NK^3)$ for K iterations. For very large dictionaries and very sparse representation, the proxy step is the dominating term, which grows linearly with dictionary size.

C. Speeding up OMP

A number of techniques have been devoted to speed up different steps of OMP. For problems in high-dimensions, i.e.
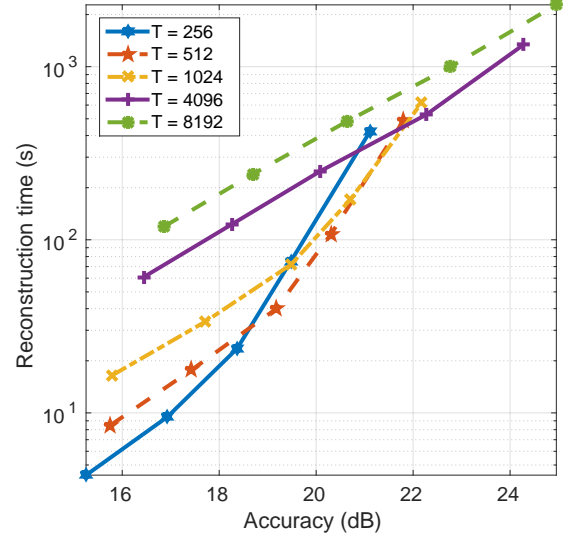

Fig. 2: Time versus accuracy for varying dictionary size when denoising $8 \times 8 \times 32$ video patches. Each curve was generated by varying the sparsity level, K, from 8 to 64 in multiples of 2. We observe that it is computationally beneficial to use a dictionary with a larger number of atoms at a smaller sparsity level as opposed to a smaller dictionary at a higher sparsity.

large values of N, one approach is to project to a lower dimension by obtaining random projections of the dictionary [17]. Specifically, as opposed to the objective $\|\mathbf{x} - D\mathbf{s}\|_2$, we minimize $\|\Phi\mathbf{x} - \Phi D\mathbf{s}\|_2$ where $\Phi \in \mathbb{R}^{M \times N}$, $M < N$, is a random matrix that preserves the geometry of the problem thereby allowing us to perform all computations in an M-dimensional space. In the context of high-dimensional data, it is typical to have dictionaries with very large number of atoms, i.e $T \gg N$, and in such a setting, the proxy step becomes a bottle neck. The need for large dictionaries is driven by a requirement for higher accuracy of reconstruction. Empirically, larger dictionaries are needed for higher accuracy, which is evident from the time vs accuracy plot in Figure 2 for denoising of videos.

D. Multi-scale approaches for sparse approximation

Various methods have been proposed to speed up sparse approximation by imposing structure on the coefficients. Such methods employ a tree-like arrangement of the sparse coefficients which give it a logarithmic complexity improvement.

One approach is by using approximate nearest neighbors and shallow-tree based matching to speed up the proxy step [18], [19]. Instead of searching across all elements of the dictionary, the dictionary is arranged into a shallow tree for fast search. In certain conditions, an $\mathcal{O}(\log(T))$ search complexity can be obtained. However, this results in a reduction in accuracy, as the closest dictionary atom is computed through approximate nearest neighbor method.

Another approach is to restrict the search space by imposing a tree structure on sparse coefficients [20]. Complexity of the proxy step would then reduce to $\mathcal{O}(\log(T))$. Restricting search space through prediction has been explored in [19] for approximation by chirplet atoms. Here, the method first finds

an approximation to the input signal through a gabor atom and, subsequently, the scale and chirp parameters are optimized locally. Though such methods provide significant speedups, their usage is restricted to signals with known structure, such as wavelets for images or hierarchical structure of chirplet atoms for sound.

E. Dictionary learning

For signal classes that have no obvious sparsifying transforms, a promising approach is to *learn* a dictionary that provides sparse representations for the specific signal class of interest. Field and Olshausen [8], in a seminal contribution, showed that patches of natural images were sparsified by a dictionary containing Gabor-like atoms — this provided a connection between sparse coding and the receptor fields in the visual cortex. More recently, Aharon et al. [9] proposed the "K-SVD" algorithm which can be viewed as an extension of the k-means clustering algorithm for dictionary learning. Given a collection of training data $X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_Q], \mathbf{x}_i \in \mathbb{R}^N$, K-SVD aims to learn a dictionary $D \in \mathbb{R}^{N \times T}$ such that $X \approx D[\mathbf{s}_1, \ldots, \mathbf{s}_Q]$ with each $\mathbf{s}_k$ being K-sparse. This was one of the first forays into learning good dictionaries for sparse representation. However, with increasing complexity and signal dimension, larger dictionaries are needed, which requires larger computation time.

F. Multi-scale dictionary models

Learning dictionary atoms that are innately clustered is an intuitive way of speeding up the approximation process with large dictionaries. Particularly for visual signals, clustering by incorporating scale or spatial complexity of the signal has been explored before. Jayaraman et al. [21] learn dictionaries by a multi-level representation of image patches where simple patches are captured in the early stages while more complex textures are only resolved at the higher levels. This provides speedups when solving sparse approximation problems since patches that occur more often are captured at the earlier levels. While speedups are constant when compared to a dictionary of the same size, it does not scale up well to high dimensional signals. Similar to this work, we propose multiple levels of representation across scales that captures complex patterns at finer scales while also incorporating a predictive framework.

Imposing a tree structure on sparse coefficients to learn dictionaries has been explored in the context of images. Jenatthon et al. [22] present a hierarchical dictionary learning mechanism, where they impose a tree structure on the sparsity, which forces the dictionary atoms to cluster like a tree. Though it does give higher accuracy of reconstruction, not much has been said about the speed up obtained. Mairal et al. [23] learn a dictionary based on quad-tree models, where each patch is further sub-divided into four non-overlapping patches. While this method gives better accuracy, the algorithm is very slow, as it involves approximations of successive decomposition of a big image patch into smaller image patches. None of the multi-scale learning algorithms exploit the cross-scale structure underlying visual signals.

G. Compressive sensing (CS)

An application of sparse representations is in CS where signals are sensed from far-fewer measurements than their dimensionality [24]. CS relies on low-dimensional representations for the sensed signal such as sparsity under a transform or a dictionary. There is a rich body of work on applying CS to imaging of visual signals including images [25], [26] videos [10], [27], [28] and light fields [11], [29]. Most relevant to our paper is the video CS work of Hitomi et al. [10] where a sparsifying dictionary is used on video patches to recover high-speed videos from low-frame rate sensors. Hitomi et al. also demonstrated the accuracy enabled by very large dictionaries; specifically, they obtained remarkable results with a dictionary of $T = 100,000$ atoms for video patches of dimension $N = 7 \times 7 \times 36 = 1764$. However, it is reported that the recovery of 36 frames of videos took more than an hour with a $100,000$ atom dictionary. Clearly, there is a need for faster recovery techniques.

H. Wavelet-tree model

Our proposed method is inspired by multi-resolution representations and tree-models enabled by wavelets. Baraniuk [4] showed that the non-zero wavelet coefficients form a rooted sub-tree for signals that have trends (smooth variations) and anomalies (edges and discontinuities). Hence, piecewise-smooth signals enjoy a sparse representation with a structured support pattern with the non-zero wavelet coefficients forming a rooted sub-tree. Similar properties have also been shown for 2D images under the separable Haar basis [5]. However, in spite of these elegant results for images, there are no obvious sparsifying bases for higher-dimensional visual signals like videos and light field images. To address this, we build cross-scale predictive models, similar to the wavelet tree model, by replacing a basis with an over-complete dictionary that is capable of providing a sparse and predictive representation for a wide class of signals.

III. CROSS-SCALE PREDICTIVE MODELS

A. Proposed model

The proposed signal model extends the notion of multi-resolution representation of signals beyond images. Given a signal f, we can represent it in the multi-resolution framework [30] as:

$$A^{2^j} f(x) = \sum_{k=1}^{N_{2^j}} \lambda_k \phi_k^{2^j}(x) = \Phi^{2^j} \Lambda^{2^j},$$

where A^{2^j} is the projection operator to the 2^j scale space, and $\{\phi_k^{2^j}(x)\}$ form a wavelet basis at 2^j scale. For piecewise constant signals like images, the wavelet coefficients form a rooted sub-tree. While it is hard to find such analytical bases for an arbitrary signal class, we can instead retain the multi-resolution framework but replace the bases Φ^{2^j} with overcomplete dictionaries. Hence, we propose a signal model that predicts the support of a signal across scales (see Figure 3). We present our model with two-scale scenario for ease of understanding. Given a collection of signals, $\mathcal{X} \subset \mathbb{R}^N$, our
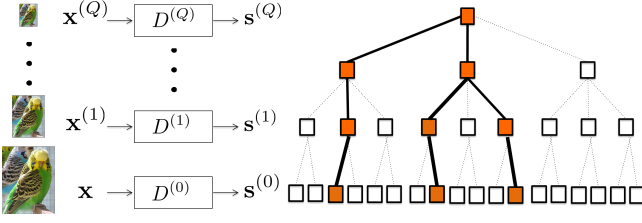
Fig. 3: Proposed cross-scale signal model with sparse coefficients across scales forming a rooted subtree. We analyze the signal $\mathbf{x}$ at multiple scales such that $\mathbf{x}^{(k)}$ is obtained by downsampling it successively k-times. At the k-th scale, we learn a dictionary $D^{(k)}$ such that sparsifies the downsamped signal $\mathbf{x}^{(k)}$, i.e, $\mathbf{x}^{(k)} = D^{(k)}\mathbf{s}^{(k)}$. We arrange the sparse coefficients $\{\mathbf{s}^{(k)}\}$ onto a tree and enforce the cross-scale prediction property as follows: a child atom can take non-zero values only if its parent is non-zero.

proposed signal model consists of two sparsifying dictionaries, $D_{\text{high}} \in \mathbb{R}^{N \times T_{\text{high}}}$ and $D_{\text{low}} \in \mathbb{R}^{N_{\text{low}} \times T_{\text{low}}}$, that satisfy the following three properties.

- *Sparse approximation at the finer scale.* A signal $\mathbf{x} \in \mathcal{X}$ enjoys a K_{high}-sparse representation in D_{high}, i.e, $\mathbf{x} \approx D_{\text{high}}\mathbf{s}_{\text{high}}$ with $\|\mathbf{s}_{\text{high}}\|_0 \leq K_{\text{high}}$.
- *Sparse approximation at the coarser scale.* Given $\mathbf{x} \in \mathcal{X}$ and a downsampling operator $W : \mathbb{R}^N \mapsto \mathbb{R}^{N_{\text{low}}}$, the downsampled signal $\mathbf{x}_{\text{low}} = W\mathbf{x}$ enjoys a sparse representation in D_{low}, i.e., $\mathbf{x}_{\text{low}} \approx D_{\text{low}}\mathbf{s}_{\text{low}}$ with $\|\mathbf{s}_{\text{low}}\|_0 \leq K_{\text{low}}$. The downsampling operator W is domain specific.
- *Cross-scale prediction.* The support of $\mathbf{s}_{\text{high}}$ is constrained by the support of $\mathbf{s}_{\text{low}}$; specifically, $\Omega_{\mathbf{s}_{\text{high}}} \subset f(\Omega_{\mathbf{s}_{\text{low}}})$, where the mapping $f(\cdot)$ is known a priori.

We make a few observations.

Observation 1. $T_{\text{high}} \gg T_{\text{low}}$ since $N_{\text{high}} \gg N_{\text{low}}$. With the increase of dimension of the signal, more complex patterns emerge which require larger number of redundant elements. Empirically we found that the number of atoms in a dictionary increases super linearly with increasing dimension of the signal for a given approximation accuracy (see Figure 2).

Observation 2. Recall that the computational time of OMP is proportional to the number of atoms in the dictionary since, at each iteration of the algorithm, we need to compute the inner product between the residue and the atoms in the dictionary. If we can constrain the search space by constraining the number of atoms, then we can obtain computational speedups.

The proposed model obtains speedups by first solving a sparse approximation problem at the coarse scale and subsequently exploiting the cross-scale prediction property to constrain the support at the finer scale. The source of the speedups relies on two intuitive ideas: first, solving a sparse approximation problem for a problem with fewer atoms (and in a smaller dimension) is faster due to OMP's runtime being linear in the number of atoms of the dictionary used [31]; and second, if we knew the support of $\mathbf{s}_{\text{low}}$, then we can simply discard all atoms in D_{high} that do not belong to $f(\Omega_{\mathbf{s}_{\text{low}}})$ since the support of $\mathbf{s}_{\text{high}}$ is guaranteed to lie within $f(\Omega_{\mathbf{s}_{\text{low}}})$.

B. Cross-scale mapping

We propose the following strategy for the cross-scale mapping f. Let $Q = T_{\text{high}}/T_{\text{low}}$ (assuming T_{high} and T_{low} are chosen to ensure Q is an integer). The cross-scale prediction map is defined using this simple rule.

$$i \in \Omega_{\mathbf{s}_{\text{low}}} \implies (i-1)Q + \{1, 2, \ldots, Q\} \subset f(\Omega_{\mathbf{s}_{\text{low}}})$$

Each element of the support $\Omega_{\mathbf{s}_{\text{low}}}$ in the coarser scale controls the inclusion/exclusion of a *non-overlapping block* of locations for the sparse vector in the finer scale. As a consequence, the cardinality of $f(\Omega_{\mathbf{s}_{\text{low}}})$ is QK_{low}.

C. Solving inverse problems under the proposed signal model.

We now detail the procedure for solving a sparse approximation problem using the proposed signal model (see Figure 3). Specifically, we seek to recover $\mathbf{x} \in \mathcal{X}$ from a set of linear measurements $\mathbf{y} \in \mathbb{R}^M$ of the form

$$\mathbf{y} = \Phi\mathbf{x} + \mathbf{e} = \Phi D_{\text{high}}\mathbf{s}_{\text{high}} + \mathbf{e},$$

where $\Phi \in \mathbb{R}^{M \times N}$ is the measurement matrix and $\mathbf{e}$ is the measurement noise. As indicated earlier, we obtain $\mathbf{s}_{\text{high}}$ using a two-step procedure.

Step1 — Sparse approximation at the coarse scale. We first solve the following sparse approximation problem:

$$(P_{\text{low}}) \qquad \widehat{\mathbf{s}}_{\text{low}} = \arg\min_{\mathbf{s}_{\text{low}}} \|\mathbf{y} - \Phi U D_{\text{low}}\mathbf{s}_{\text{low}}\|_2$$

$$\text{s.t.} \qquad \|\mathbf{s}_{\text{low}}\|_0 \leq K_{\text{low}}.$$

Here, $U : \mathbb{R}^M \mapsto \mathbb{R}^N$ is an up-sampling operator such that WU is an identity map on $\mathbb{R}^{N_{\text{low}}}$. In all our experiments, we used a uniform downsampler and a nearest-neighbour up sampler specific to the domain of the signal. This step recovers a low-resolution approximation to the signal, $\mathbf{x}_{\text{low}} = D_{\text{low}}\widehat{\mathbf{s}}_{\text{low}}$.

Step 2 — Sparse approximation at the finer scale. Armed with the support $\widehat{\Omega} = \Omega_{\widehat{\mathbf{s}}_{\text{low}}}$, we can solve for $\mathbf{s}_{\text{high}}$ by solving:

$$(P_{\text{high}}) \qquad (\widehat{\mathbf{s}}_{\text{high}})_{|f(\Omega)} = \arg\min_{\alpha} \|\mathbf{y} - \Phi(D_{\text{high}})_{|\Omega}\alpha\|_2$$

$$\text{s.t.} \qquad \|\alpha\|_0 \leq K_{\text{high}}.$$

The sparse approximation problems in both steps are solved using OMP. The proposed mapping across scales for the sparse support forms a zero tree, where a coefficient is zero if the corresponding coefficient at coarser scale is zero. Hence we refer to our algorithm as zero tree OMP. Algorithm 2 outlines the zero tree OMP procedure.

D. Theoretical speedup.

We provide expressions for the expected speedups over traditional single-scale OMP. Since any analysis of speedup has to account for the complexity of implementing Φ, we consider the denoising problem where Φ is the identity matrix.

Let $C(N, T, K)$ be the amount of time required to solve a sparse-approximation problem using OMP for a dictionary of size $N \times T$ and sparsity level K. Hence, obtaining $\mathbf{s}_{\text{high}}$ directly from $\mathbf{x}$ would require $C(N, T_{\text{high}}, K_{\text{high}})$ computations. In contrast, our proposed two-step solution using cross-scale prediction has a computational cost of $C(N_{\text{high}}, T_{\text{low}}, K_{\text{low}}) + C(N, QK_{\text{low}}, K_{\text{high}})$.

To compute the dependence of $C(N, T, K)$ on N, T, and K, recall that for each iteration in the OMP algorithm, we need

Algorithm 2 Zero tree OMP

Require: $\mathbf{x}$, D_{low}, D_{high}, K_{low}, K_{high}, W

    $\mathbf{x}_{\text{low}} \leftarrow W\mathbf{x}$

    $r_{\text{low}} \leftarrow \mathbf{x}_{\text{low}}$

    $\Omega_{\text{low}} \leftarrow \phi$

    $\alpha_{\text{low}} \leftarrow \phi$

    for $n = 1$ to K_{low} **do**

        $k \leftarrow \arg\max_i |\langle (\mathbf{d}_{\text{low}})_i, r_{\text{low}} \rangle|$

        $\Omega_{\text{low}} \leftarrow \Omega_{\text{low}} \bigcup k$

        $\alpha_{\text{low}} \leftarrow \arg\min_\beta \|\mathbf{x}_{\text{low}} - (D_{\text{low}})_{|\Omega}\beta\|^2$

        $r_{\text{low}} \leftarrow \mathbf{x}_{\text{low}} - (D_{\text{low}})_{|\Omega}\alpha_{\text{low}}$

    end for

    $\hat{D} = (D_{\text{high}})_{|f(\Omega_{\text{low}})}$

    $r_{\text{high}} \leftarrow \mathbf{x}_{\text{high}}$

    $\Omega_{\text{high}} \leftarrow \phi$

    $\alpha_{\text{high}} \leftarrow \phi$

    for $n = 1$ to K_{high} **do**

        $k \leftarrow \arg\max_i |\langle \hat{\mathbf{d}}_i, r_{\text{high}} \rangle|$

        $\Omega_{\text{high}} \leftarrow \Omega_{\text{high}} \bigcup k$

        $\alpha_{\text{high}} \leftarrow \arg\min_\beta \|\mathbf{x}_{\text{high}} - \hat{D}_{|\Omega}\beta\|^2$

        $r_{\text{high}} \leftarrow \mathbf{x}_{\text{high}} - \hat{D}_{|\Omega}\alpha_{\text{high}}$

    end for

    return Ω_{high}, α_{high}

(a) Image denoising   (b) Image inpainting

Fig. 4: Effect of $Q = T_{\text{high}}/T_{\text{low}}$ on approximation accuracy for various signal dimensions and dictionary sizes. For 8×8 patch, the optimal Q is 4, while that for 16×16 is 16 for $T_{\text{low}} = 256$ and 8 for $T_{\text{low}} = 512$. Between the two dictionary sizes, $T_{\text{low}} = 256$ gives better performance for 16×16 patches. Metrics such as these may be used for finding the optimal Q and T_{low}.

OMP by zero-tree OMP, the overall time taken for each iteration reduces, thus speeding up the learning of dictionaries.

$\mathcal{O}(NT)$ operations [31] for finding inner product between the residue and the dictionary atoms, $\mathcal{O}(T)$ operations to find the maximally aligned vector and $\mathcal{O}(K^3 + K^2 N)$ operations for the least-squares step. Thus,

$$C(N, T, K) = \mathcal{O}(NTK + TK + K^4 + K^3 N).$$

For dictionaries with a large number of atoms, i.e., large T, and small values for sparsity level K, the linear dependence on T dominates the total computation time. Hence, the speedup provided by our algorithm is approximately $T_{\text{high}}/(T_{\text{low}} + K_{\text{low}}Q)$.

E. Learning cross-scale sparse models.

We learn the dictionaries $(D_{\text{high}}, D_{\text{low}})$ with a simple modification to the K-SVD algorithm.

Inputs. The inputs to the learning/training phases are the training dataset $X = [\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_n]$ and the values for the parameters $K_{\text{high}}, K_{\text{low}}, T_{\text{high}}$, and T_{low}.

Step 1 — Learning D_{low}. We learn the coarse-scale dictionary D_{low} by applying K-SVD to downsampled training dataset $X_{\text{low}} = [W\mathbf{x}_1, W\mathbf{x}_2, \ldots W\mathbf{x}_n]$. A by-product of learning the dictionary D_{low} are the supports $\{\Omega_{s_{\text{low}}, k_{\text{low}}}\}$ of sparse approximations of the downsampled training dataset.

Step 2 — Learning D_{high}. We learn the fine-scale dictionary $D_{\text{high}} = [\mathbf{d}_1, \ldots, \mathbf{d}_{T_{\text{high}}}]$ by solving

$$\min_{D_{\text{high}}, S_{\text{high}}} \|Y - D_{\text{high}}S_{\text{high}}\|_F \quad \text{s.t.} \quad \|\mathbf{d}_k\|_2 = 1,$$

$$\text{support}(\mathbf{s}_{\text{high}}) \subset f(\Omega_{\text{low}, k})$$

The above optimization problem can be solved simply by modifying the sparse approximation step of K-SVD to restrict the support appropriately.

As a consequence of speed up in approximation step, dictionary learning by proposed method is also faster. Recall that K-SVD alternates between dictionary learning and sparse approximation. Since the modified K-SVD algorithm replaces
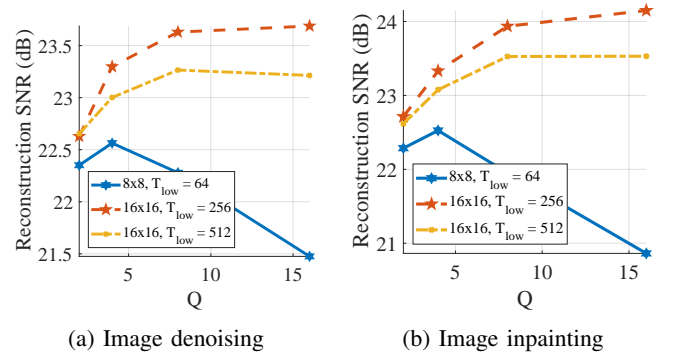
F. Parameter selection

The design parameters in the two scale dictionary training are $K_{\text{low}}, K_{\text{high}}, T_{\text{low}}$, and T_{high}. K_{low} can be chosen to fine tune the accuracy at lower scales. For compressive sensing purposes, lower sparsity promises better reconstruction results. Hence a small K_{low} gives better results. We found that K_{low} in range of $2-4$ worked well. K_{high} should be greater than or equal to K_{low}, as at least one atom corresponding to the low resolution atom will be picked.

The parameter $Q = \frac{T_{\text{high}}}{T_{\text{low}}}$ can be chosen by cross-validation. As an example, cross-validation may be performed with denoising as a test metric. The value of Q that gives highest reconstruction accuracy can then be chosen as the optimal Q. To illustrate this, we trained dictionaries for various values of T_{low} and Q for image patches of various sizes, and tested them for denoising and inpainting for the "peppers" image. Figure 4 shows a comparison of approximation accuracy for various signal dimensions with different dictionary values as a function of Q. For 8×8 image patches, $Q = 4$ gives best results, whereas it is 16 for 16×16 patches and 8 for 24×24 patches. Since retraining dictionaries for each value of Q is a time consuming process, T_{low} and T_{high} were chosen as would be appropriate for the signal dimension, N_{low} and N_{high} respectively.

G. Initialization of dictionary

Since the dictionary learning objective as well as the multi-scale dictionary learning objective are non-convex, the solution obtained depends on the initialization. Elad et al. [32] proposed certain initialization and update heuristics which ensured good results. Along the same lines, we propose the following heuristics:
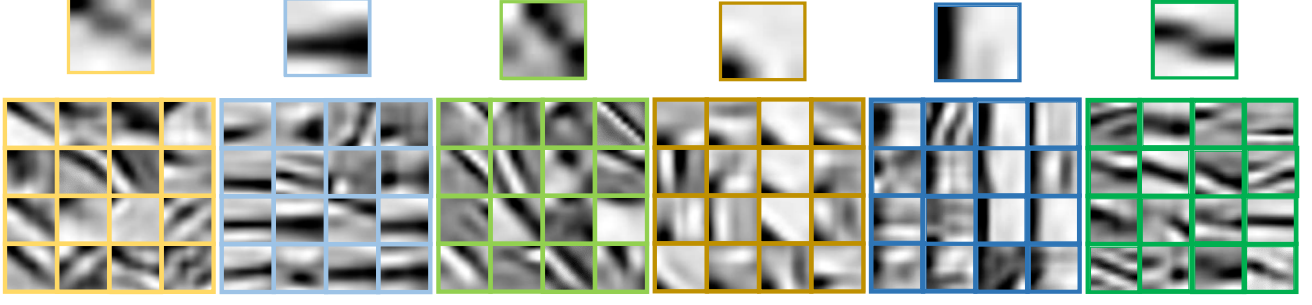
Fig. 5: Visualization of select low-resolution atoms and their corresponding atoms in the high resolution dictionary; top – low resolution atoms, scaled up to show features clearly; bottom – corresponding high resolution atoms per each low resolution atom. By restricting the support set of higher resolution approximation, our method learns child atoms similar to parent atom.
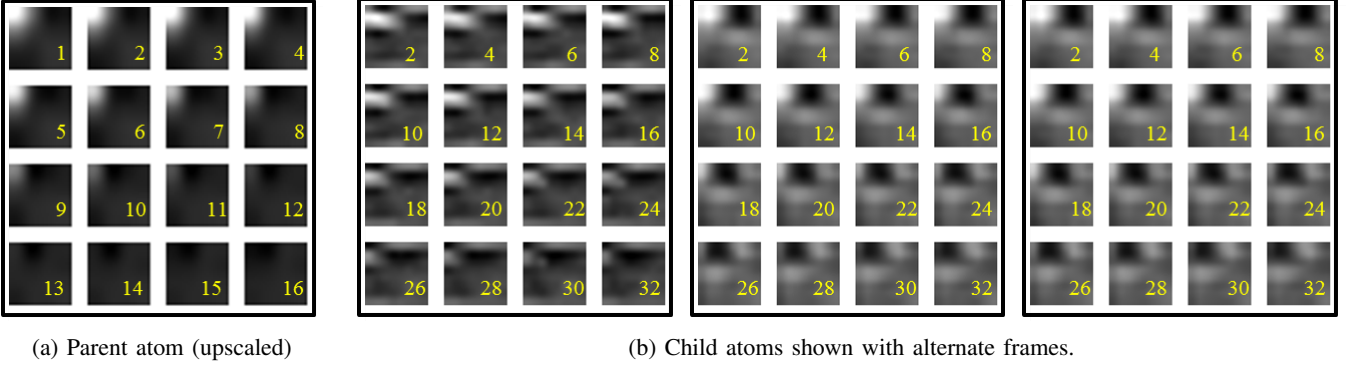

(a) Parent atom (upscaled) (b) Child atoms shown with alternate frames.

Fig. 6: Visualization of frames of a $8 \times 8 \times 32$ parent and select children atoms. Notice that the child atoms have similar motion pattern to the parent atom, with added spatial details.
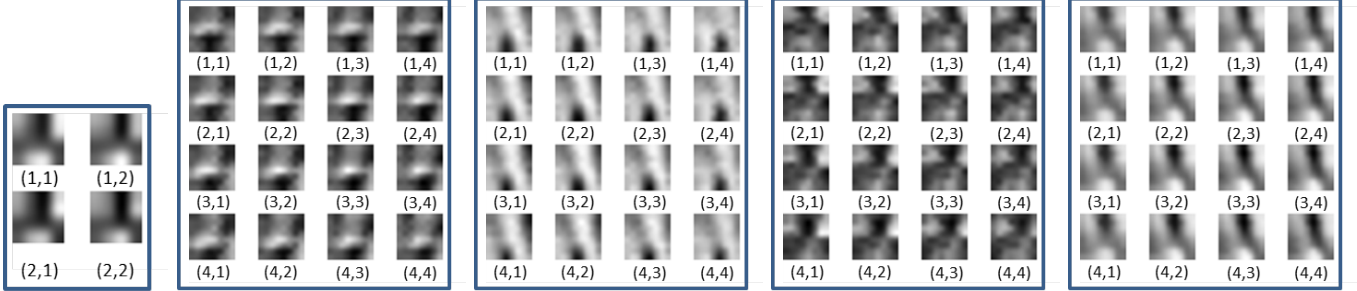

Fig. 7: Visualization of a $4 \times 4 \times 2 \times 2$ parent atom (left) and frames of select $8 \times 8 \times 4 \times 4$ high resolution child atoms. The low resolution atoms have been scaled up by a factor of 2 to show features clearly. The various sub-aperture views in child atoms are similar to the parent atoms but with added spatial details.

1) The lowest resolution dictionary may be initialized as proposed in [32]. In our experiments, we initialized by picking T_{low} training patches randomly.
2) For initializing higher resolution dictionaries, we use low resolution dictionary information. Let D_{low} be output of the first step of multi-scale dictionary training. Let $\Omega_k = \{i : W\mathbf{x}_j \approx \alpha_{j,k}\mathbf{d}_{\text{low},k} + r_j \quad \forall j = 1,2,...,n\}$. Then, $\mathbf{d}_{\text{high},j}, \forall j \in f(k)$ is randomly initialized from the sub training samples, $X_{|\Omega_k}$.
3) An unused atom from the lowest resolution dictionary may be replaced by the least represented training sample, as proposed in [32].
4) Let Ω_k be as defined above. Then an unused atom in

a higher resolution dictionary may be replaced from the least represented training sampling in X_{Ω_k}.

Figures 5, 6, and 7 show examples of the learnt low-resolution atoms and the corresponding high-resolution atoms for images, videos and light fields. Observe that constraining the sparse support of the high-resolution approximation alone learns patches which are very similar in appearance to the low-resolution patches, which supports our proposed signal model.

IV. EXPERIMENTAL RESULTS

A. Simulation details

To validate our signal model, we show that our signal model performs as good as a large dictionary with runtimes compared
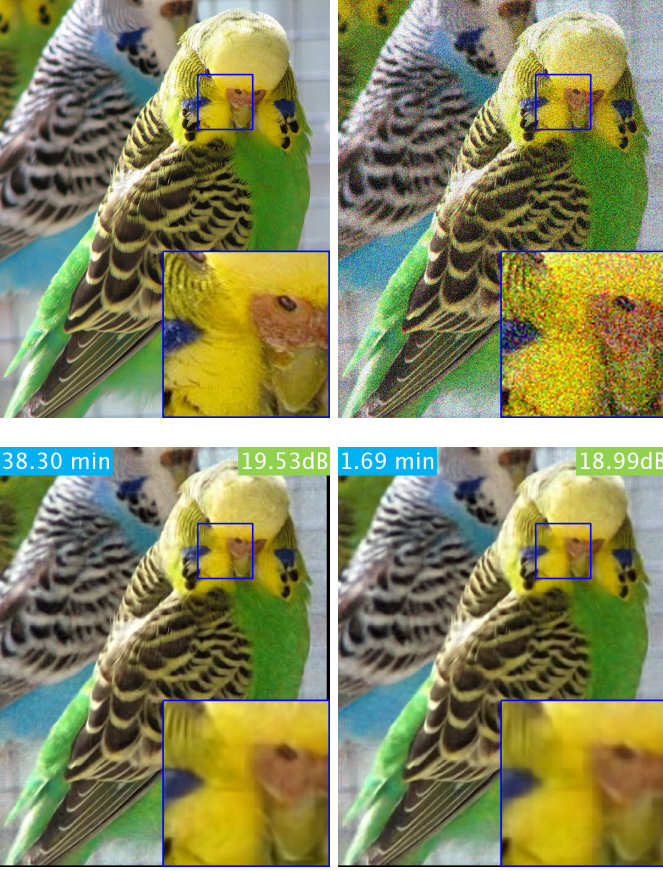
Fig. 8: Visualization of results for denoising the "Budgerier" image. Clockwise from top left, original image, noisy image with SNR of 10dB, recovered image using proposed method, and recovered image using K-SVD learned dictionary. We obtain a speedup of $22\times$ with less than 1dB reduction in accuracy.

to that of a small dictionary. We trained dictionaries over various classes of visual signals to emphasize the ubiquity of our signal model. Comparisons were made against a small dictionary with (1) N_{low} atoms, (2) a large dictionary with N_{high} atoms, (3) our proposed multiscale dictionary with N_{low} low resolution and N_{high} high resolution atoms, and (4) a N_{high} multi-level dictionary with N_{low} levels, as proposed by Jayaraman et al. [21]. We quantify the approximation accuracy using recovered SNR that is defined as follows: given a signal $\mathbf{x}$ and its estimate $\widehat{\mathbf{x}}$, SNR $= 20\log_{10}(\|\mathbf{x}\|/\|\mathbf{x} - \widehat{\mathbf{x}}\|)$.

B. Images

We trained dictionaries with $N_{\text{low}} = 512$ and $N_{\text{high}} = 8192$ on 24×24 image patches and downscaled patches of dimension 12×12.

Figure 1 shows demosaicing of the Bayer pattern using a large single scale dictionary and our proposed method. We trained an 8192 atom high resolution dictionary on 24×24 Kodak True color RGB images [33] and 512 atom low resolution dictionary on the patches downscaled to 12×12. We compare this against 8192 atom single scale dictionary. It took 16 minutes for the single scale with an approximation accuracy
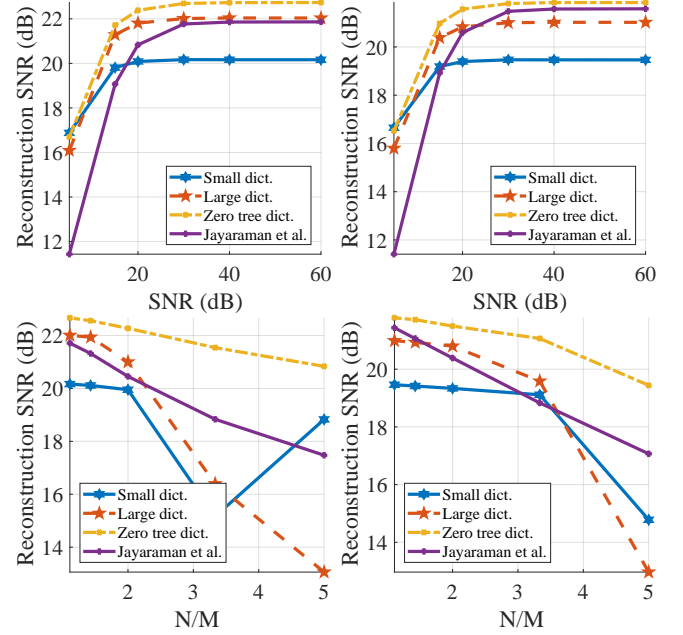

Fig. 9: Performance on (top-row) denoising with additive Gaussian noise and (bottom-rown) inpainting with randomly removed pixels for two images — (left column) peppers and (right) budgerier. Here, N/M represents the number of unknowns per each known. See Section IV-A for details about each dictionary. The proposed method provides better reconstruction accuracy over competing methods.

of 18.45dB, whereas only 1.5 minutes with an approximation accuracy of 18.43dB for the two scale dictionary.

Figure 8 shows image denoising at an SNR of 10dB. We perform denoising with the trained RGB dictionaries of 24×24 patch and with a patch overlap of 18 pixels. With hardly any reduction in accuracy, our method performs $22\times$ faster. Figure 9 compares the performance of various dictionaries for denoising and CS tasks for two representative images. For CS, we retained known pixel values only at a fraction of the locations and recovered the complete image. For both the cases, our dictionary outperforms other methods. Speed ups obtained for denoising and CS is summarized in Table I. With 1dB or less loss in accuracy, our method offers significant speed ups for all image processing tasks.

It is worth mentioning that BM3D [34], [35], one of the classical image denoising techniques that uses non-local statistics, provides exceptional denoising results; typically, at 15 dB measurement noise BM3D outperforms most sparse optimization-based denoisers — including the proposed method — by 9 dB or so. However, the run times associated with BM3D are often longer than our approach. Further, it is also worth noting that the proposed idea as well as most dictionary-based representations are designed towards solving general linear inverse problems that go beyond denoising.

C. Videos

We trained dictionaries with $N_{\text{low}} = 512$ and $N_{\text{high}} = 8192$ on $8 \times 8 \times 32$ video patches and downscaled video patches of dimension $4 \times 4 \times 16$. We show empirically that our

Signal Class	N	N_{low}	T_{low}	T_{high}	K_{low}	K_{high}	Denoising			N/M	Inpainting		
							Speedup	Single scale (dB)	Multi-scale (dB)		Speedup	Single scale (dB)	Multi-scale (dB)
Images	8x8	4x4	64	512	4	6	3.86	19.6	18.6	5	1.61	18.52	18.3
	24x24	12x12	512	8192	4	6	40.72	21.3	19.9	5	16.12	20.8	19.6
Videos	8x8x32	4x4x16	512	8192	4	6	23.00	19.36	20.14	8	7.9	16.63	16.32
Light field	4x4x8x8	2x2x4x4	1024	8192	4	6	66.34	23.13	26.32	2	19.07	15.9	17.75

TABLE I: Table with speed up and accuracy for denoising and CS with various class of signal. Denoising was performed with 15dB SNR conditions. N/M represents the number of unknowns per each known variable.
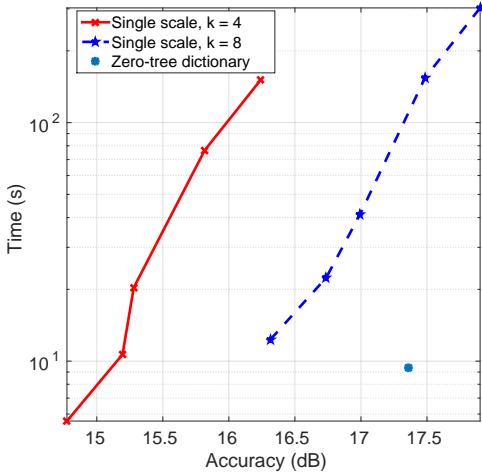

Fig. 10: Time vs accuracy for denoising of videos at $15dB$ input noise, with single scale dictionary and proposed multi-scale dictionary. Dictionaries of sizes 256, 512, 1024, 4096 and 8192 are compared against a zero tree dictionary of 8192 atoms of high resolution and 512 atoms of low resolution, with $K_{\text{low}} = 6$ and $K_{\text{high}} = 8$. At high approximation accuracies, our method outperforms large dictionaries in run-time time.

signal model outperforms single scale dictionaries in terms of speed and accuracy. Figure 10 show the comparison of single scale dictionaries of various size and zero-tree dictionary for denoising of videos. For the same time of approximation, our method gives the highest accuracy. Put it another way, for the same accuracy, our signal model takes the least time.

Figure 11 shows the performance of various dictionaries for denoising and CS tasks and Figure 12 show results for CS of videos. For CS, we combined multiple frames into a coded image, as proposed by Hitomi et al. [10]. Speedup in denoising was $20\times$ while that for CS was between $5\times$ and $15\times$, depending on the number of measurements. Performance for video denoising and CS has been summarized in Table I. Speedups obtained for videos is much higher than for images with less than 1dB loss in accuracy. Results are significantly better visually too, as can be seen in Figure 12, from the smoother spatial profile compared to reconstruction with single scale dictionary.
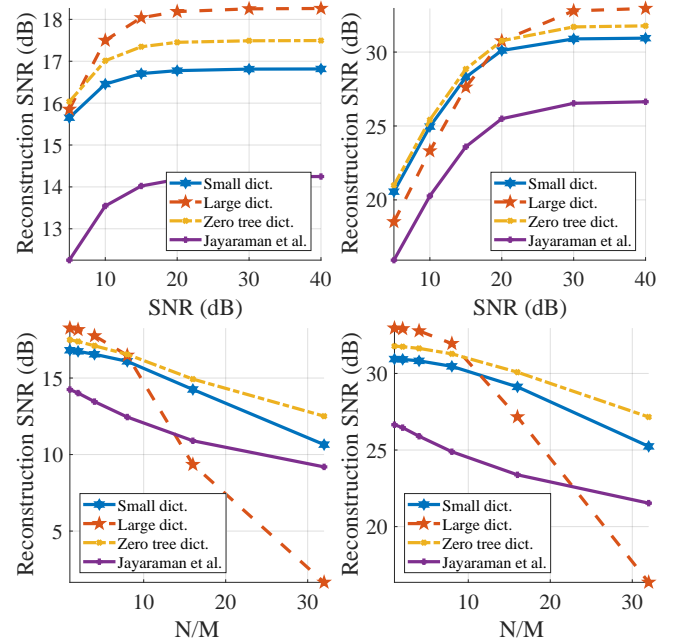

Fig. 11: Reconstruction performance on (top-row) denoising with additive Gaussian noise and (bottom) compressive sensing using the imaging architecture of Hitomi et al. [10] for two different videos — (left) sharpner and (right) egg drop. Here, N/M represents the number of frames reconstructed from each coded image.

D. Light field images

We trained dictionaries with $N_{\text{low}} = 512$ and $N_{\text{high}} = 8192$ on $4 \times 4 \times 8 \times 8$ video patches and downscaled video patches of dimension $2 \times 2 \times 4 \times 4$. Figure 13 shows the performance of various dictionaries for denoising and CS tasks and Figure 14 shows results for CS of light fields. For CS, we simulated acquisition of images with multiple coded aperture settings, proposed in [36]. Performance metrics for denoising and CS has been summarized in Table I. Our method not just offers a $19 - 60\times$ speed up, but there is an increase in reconstruction accuracy. Improved results can be observed visually in the reconstruction results of the Dragons and Buddha datasets in Figure 14 from compressive measurements.
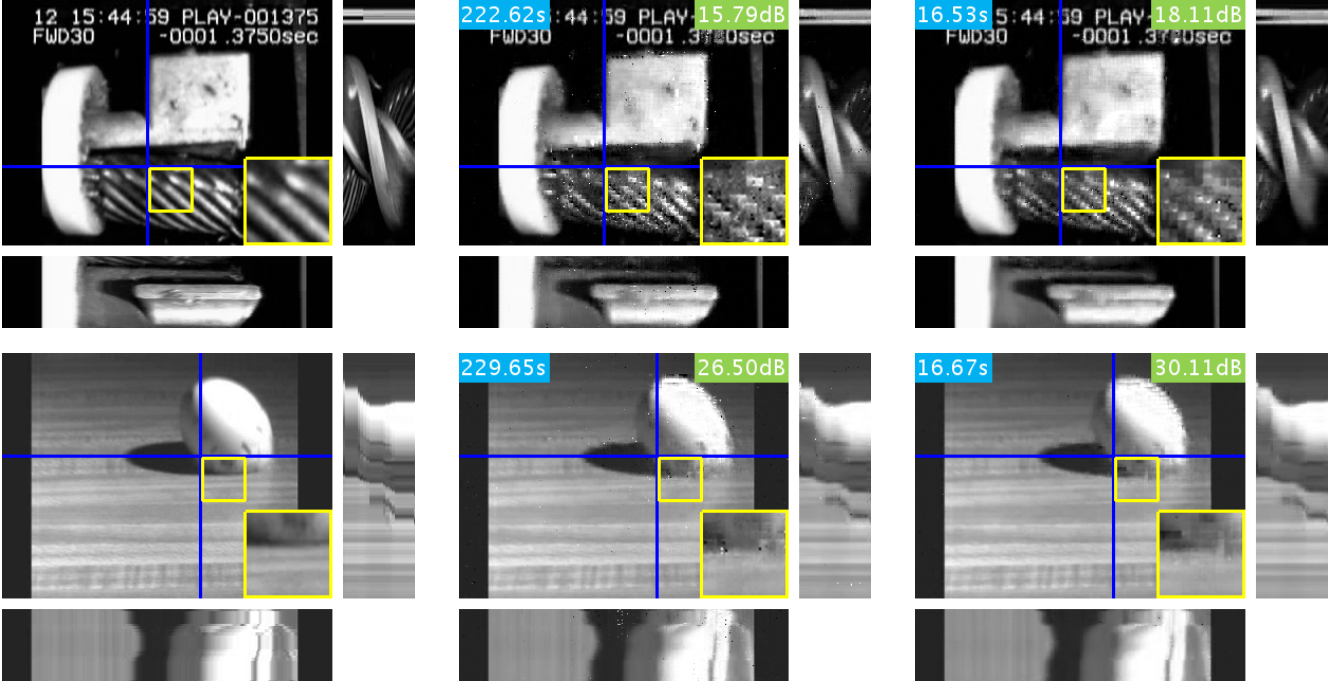
Fig. 12: Visualization of reconstructed frames for (top-row) sharpner and (bottom) egg drop videos; (columns: from left-to-right) ground truth, reconstruction with single scale dictionary; reconstruction with two scale dictionary. The proposed method provides a speedup of $14\times$ over OMP with a modest 2-4 dB improvement in reconstruction SNR.
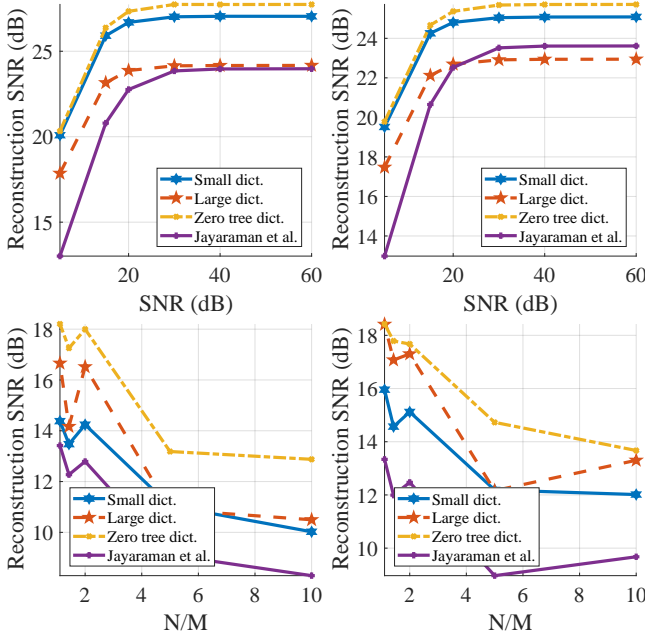

Fig. 13: Reconstruction performance on (top row) denoising and (bottom) CS with imaging architecture of Liang et al. [36] for two different light fields – (left column) Buddha and (right) Dragons. Here, N/M represents the number of sub-aperture views recovered for each coded aperture image.

E. Experiments on real data

We tested our algorithm on real data collected by Hitomi et al. [10]. A $320 \times 320 \times 18$ video was reconstructed from a 320×320 coded image. We compared reconstruction with our zero-tree dictionary and the dictionaries of 10,000 and 100,000 atoms trained by Hitomi et al. Figure 15 show the reconstruction results for a bouncing ball and a toy airplane respectively with the time taken for reconstruction. Notice that the results are visually similar while our method is faster by $6\times$ compared to the stock 10,000 atom dictionary. In case of the bouncing ball, the number "4" has been better resolved in our result as well.

F. Summary

Table II and Figures 9, 11 and 13 quantify the performance of the proposed signal model and those obtained using K-SVD for a wide range of parameters as well as signals. Across the board, we observe that the proposed framework provides accuracies that are as good as those obtained with K-SVD, but with speedups that are $4 - 10\times$ for small-sized problems and $20 - 110\times$ for larger problems. The speedups obtained are comparable to results in [18] with higher approximation accuracies for our proposed method. As a result of speed up of the sparse coding step, we also get significant speed ups during the training phase $(2 - 40\times)$ using modified K-SVD, which makes it feasible to deal with very large problems.

V. CONCLUSION AND DISCUSSIONS

We presented a signal model that enables cross scale predictability for visual signals. Our method is particularly appealing because of the simple extension to the existing OMP and K-SVD algorithms while providing significant speed ups at little or no loss in accuracy. The computational gains provided by our algorithm are especially significant for problems involving high-dimensional dictionaries with a large number
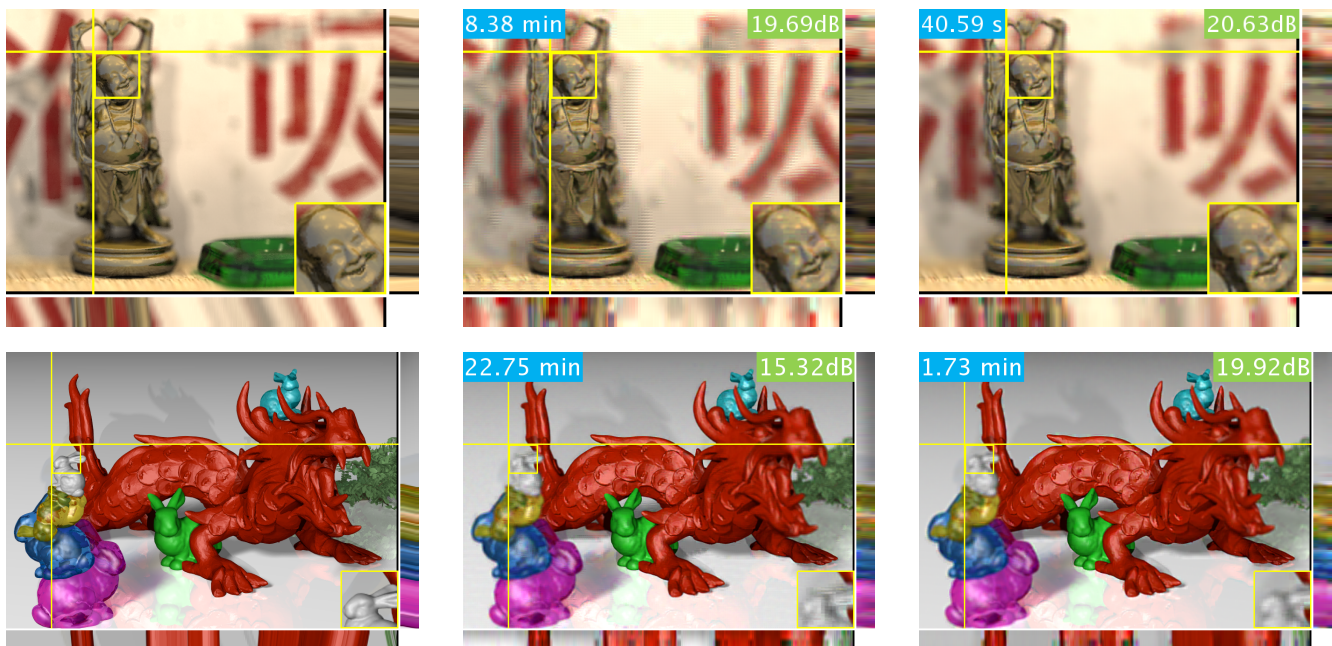
Fig. 14: Visualization of reconstructed sub-aperture views for (top-row) Buddha and (bottom) dragon light fields; (columns: from left-to-right) ground truth, reconstruction with single scale dictionary; reconstruction with two scale dictionary. The epipolar slices have been scaled up to show features clearly. The proposed method enables a $10\times$ speed up along with an increase of $1 - 4$ dB in reconstruction SNR.
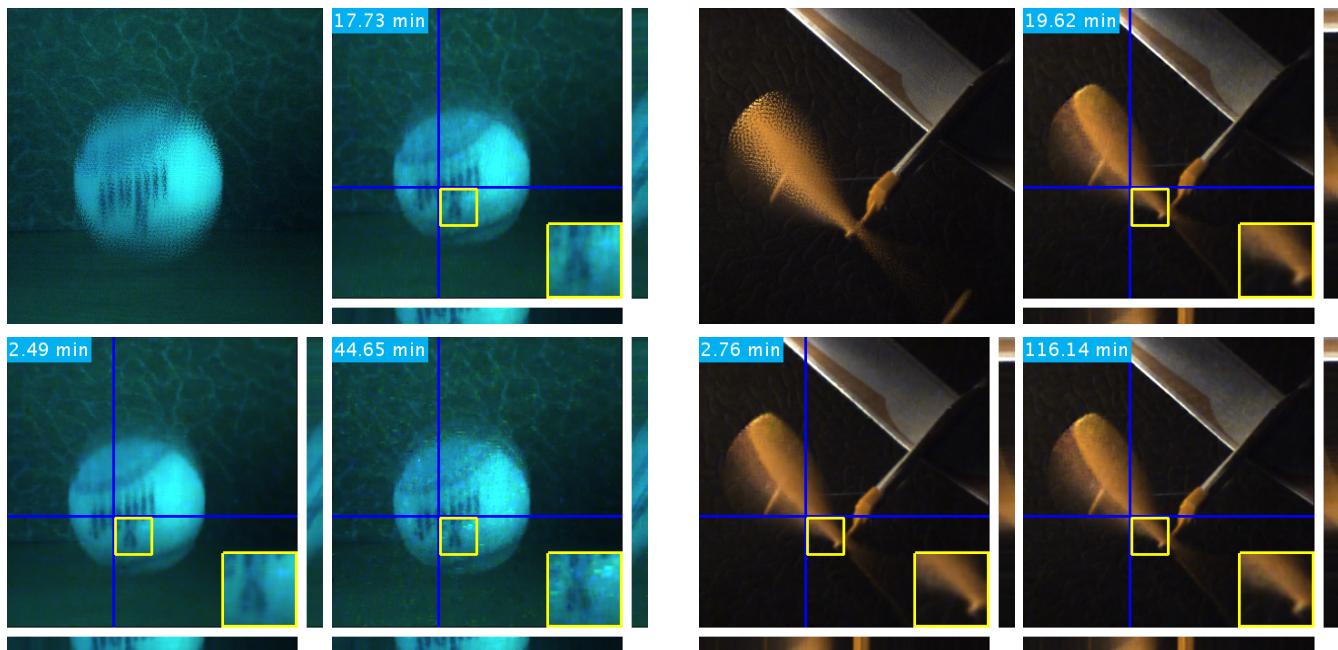

Fig. 15: Reconstruction from real data for two scenes from [10]. For each of the scenes, clockwise from top left: coded image, reconstruction with 10,000 atom dictionary, reconstruction with 100,000 atom dictionary and reconstruction with the proposed two-scale dictionary. The proposed method provides a $6\times$ speed up over the 10,000 atom dictionary at an improved visual quality — for example, the number "4" is rendered with fewer artifacts in our reconstruction.

of atoms. We also believe that the proposed cross scale predictive models can be incorporated with other structural modifications to sparse dictionaries. One such example is that of convolutional sparse coding [37]–[39], where each dictionary atom is a convolutional filter and, unlike a patch-based method, the method approximates the sparse coefficients for an entire image. Using cross scale predictive modeling on top would in principle lead to runtime speedups.

Signal Class	N	N_{low}	T_{low}	T_{high}	K_{low}	K_{high}	Speedup	Single scale (dB)	Multi-scale (dB)
Images	8x8	4x4	64	1024	8	8	4.10	21.98	20.67
	24x24x3	12x12x3	512	8192	8	8	22.6	20.57	19.64
Videos	8x8x16	4x4x8	512	8192	16	16	15.87	24.09	22.62
	8x8x16	4x4x8	512	8192	14	16	15.80	24.09	22.75
	8x8x32	4x4x16	512	8192	16	16	23.81	21.36	20.72
	8x8x16	4x4x8	512	16384	16	16	16.89	23.27	21.84
Light field Images	4x4x8x8	2x2x4x4	2048	32768	4	4	111.02	21.73	19.91
	4x4x8x8	2x2x4x4	1024	8192	4	6	66.34	23.45	23.04

Legend

N	Size of the high resolution signal
N_{low}	Size of the low resolution signal
T_{low}	Number of atoms in low resolution dictionary
T_{high}	Number of atoms in high resolution dictionary
K_{low}	Sparsity used in low resolution dictionary
K_{high}	Sparsity used in high resolution dictionary
Speed up	Ratio of time taken for single scale approximation by time taken for two scale approximation

TABLE II: Table with speed up for various dictionary sizes, patch sizes and sparsity. The speed up shown are for solving sparse approximation problems and quantify the ratio of time taken by OMP using a K-SVD learnt dictionary to zero tree OMP on the proposed model. Also shown are approximation errors on training dataset for both K-SVD and the proposed algorithm.

A. Limitations

In order to get higher accuracy of construction, the sparsity levels need to be higher than that for large scale dictionaries with the same number of atoms as the high resolution dictionary. However, this is not a major drawback, as the speedups are still significant in spite of the increased sparsity levels.

Table II shows that the model accuracy for our proposed signal model is lower than that of large dictionary. This is due to two reasons:

1) K-SVD algorithm, being a non-convex optimization framework, is very sensitive to initialization. The initialization proposed in this paper is at best a heuristic. Better results can be obtained with better initialization methods.
2) The dictionary update step for the proposed modified K-SVD algorithm runs independent of the sparse approximation step. A better approach would be to modify the dictionary update step to incorporate the interdependence of sparse coefficients.

B. Connections to super resolution using dictionaries

Roman et al. [40] learn a pair of low resolution and high resolution dictionary using the same sparsity pattern for the two dictionaries. Given a low resolution patch $\mathbf{y}_{low}$, the sparse approximation problem $\mathbf{y}_{low} \approx D_{low}\mathbf{s}$ is first solved and subsequently the low-resolution image is super resolved as $\mathbf{y} = D_{high}\mathbf{s}$. In contrast, our method requires the high resolution image as an input and uses the sparse representation of the downscaled image to predict the high resolution sparse representation. While the primary aim of [40] is image-based super resolution, our method can accommodate any inverse problem based on sparse approximation.

References

[1] E. Adelson, E. Simoncelli, and W. T. Freeman, "Pyramids and multiscale representations," *Representations and Vision, Gorea A.,(Ed.). Cambridge University Press, Cambridge*, 1991.

[2] A. Secker and D. Taubman, "Lifting-based invertible motion adaptive transform (limat) framework for highly scalable video compression," *IEEE Trans. Image Processing*, vol. 12, no. 12, pp. 1530–1542, 2003.

[3] M. J. Wainwright, E. P. Simoncelli, and A. S. Willsky, "Random cascades on wavelet trees and their use in analyzing and modeling natural images," *Appl. Comp. Harmonic Analysis*, vol. 11, no. 1, pp. 89–123, 2001.

[4] R. G. Baraniuk, "Optimal tree approximation with wavelets," in *SPIE Intl. Symp. Optical Science, Engineering, and Instrumentation*, 1999.

[5] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Processing*, vol. 41, no. 12, pp. 3445–3462, 1993.

[6] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde, "Model-based compressive sensing," *IEEE Trans. Information Theory*, vol. 56, no. 4, pp. 1982–2001, 2010.

[7] S. Deutsch, A. Averbush, and S. Dekel, "Adaptive compressed image sensing based on wavelet modeling and direct sampling," in *SAMPTA*, 2009.

[8] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by V1?" *Vision research*, vol. 37, no. 23, pp. 3311–3325, 1997.

[9] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[10] Y. Hitomi, J. Gu, M. Gupta, T. Mitsunaga, and S. K. Nayar, "Video from a single coded exposure photograph using a learned over-complete dictionary," in *IEEE Intl. Conf. Computer Vision*, 2011.

[11] K. Marwah, G. Wetzstein, Y. Bando, and R. Raskar, "Compressive light field photography using overcomplete dictionaries and optimized projections," *ACM Trans. Graphics*, vol. 32, p. 46, 2013.

[12] Z. Hui and A. C. Sankaranarayanan, "A dictionary-based approach for estimating shape and spatially-varying reflectance," in *IEEE Intl. Conf. Computational Photography (ICCP)*, 2015.

[13] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Asilomar Conf. Signals, Systems, Computers*, 1993.

[14] V. Saragadam, A. C. Sankaranarayanan, and X. Li, "Cross-scale predictive dictionaries for image and video restoration," in *IEEE Intl. Conf. Image Processing (ICIP)*, 2016.

[15] M. Elad, *Sparse and redundant representations: From theory to applications in signal and image processing*. Springer, 2010.

[16] B. K. Natarajan, "Sparse approximate solutions to linear systems," *SIAM j. computing*, vol. 24, no. 2, pp. 227–234, 1995.

[17] S. N. Vitaladevuni, P. Natarajan, and R. Prasad, "Efficient orthogonal matching pursuit using sparse random projections for scene and video classification," in *IEEE Intl. Conf. Computer Vision*, 2011.

[18] A. Ayremlou, T. Goldstein, A. Veeraraghavan, and R. G. Baraniuk, "Fast sublinear sparse representation using shallow tree matching pursuit," *arXiv preprint arXiv:1412.0680*, 2014.

[19] R. Gribonval, "Fast matching pursuit with a multiscale dictionary of gaussian chirps," *IEEE Trans. Signal Processing*, vol. 49, no. 5, 2001.

[20] C. La and M. N. Do, "Tree-based orthogonal matching pursuit algorithm for signal reconstruction," in *IEEE Intl. Conf. Image Processing*, 2006.

[21] J. J. Thiagarajan, K. N. Ramamurthy, and A. Spanias, "Learning stable multilevel dictionaries for sparse representations," *IEEE trans. neural networks and learning systems*, vol. 26, no. 9, pp. 1913–1926, 2015.

[22] R. Jenatton, J. Mairal, F. R. Bach, and G. R. Obozinski, "Proximal methods for sparse hierarchical dictionary learning," in *Intl. Conf., Machine Learning*, 2010.

[23] J. Mairal, G. Sapiro, and M. Elad, "Multiscale sparse image representation with learned dictionaries," in *IEEE Intl. Conf. Image Processing*, 2007.

[24] R. G. Baraniuk, "Compressive sensing," *IEEE Signal Processing Magazine*, vol. 24, no. 4, 2007.

[25] M. F. Duarte, M. A. Davenport, D. Takhar, J. N. Laska, T. Sun, K. E. Kelly, R. G. Baraniuk *et al.*, "Single-pixel imaging via compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, p. 83, 2008.

[26] H. Chen, M. S. Asif, A. C. Sankaranarayanan, and A. Veeraraghavan, "FPA-CS: Focal plane array-based compressive imaging in short-wave infrared," in *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 2358–2366.

[27] D. Reddy, A. Veeraraghavan, and R. Chellappa, "P2c2: Programmable pixel compressive camera for high speed imaging," in *Computer Vision and Pattern Recognition (CVPR), IEEE Conf.*, 2011, pp. 329–336.

[28] A. C. Sankaranarayanan, C. Studer, and R. G. Baraniuk, "CS-MUVI: Video compressive sensing for spatial-multiplexing cameras," in *IEEE Intl. Conf. Comp. Photography (ICCP)*, 2012, pp. 1–10.

[29] S. Tambe, A. Veeraraghavan, and A. Agrawal, "Towards motion aware light field video for dynamic scenes," in *IEEE Intl. Conf. Computer Vision*, 2013.

[30] S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE trans. pattern analysis and machine intelligence*, vol. 11, no. 7, pp. 674–693, 1989.

[31] B. Mailhé, R. Gribonval, F. Bimbot, and P. Vandergheynst, "A low complexity orthogonal matching pursuit for sparse signal approximation with shift-invariant dictionaries," in *IEEE Intl. Conf. Acoustics, Speech, Signal Processing*, 2009.

[32] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans., Image Processing*, vol. 15, no. 12, pp. 3736–3745, 2006.

[33] "Kodak lossless true color image suite," http://r0k.us/graphics/kodak/, last accessed: 2015-10-05.

[34] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Trans. image processing*, vol. 16, no. 8, pp. 2080–2095, 2007.

[35] D. Kostadin, F. Alessandro, and E. KAREN, "Video denoising by sparse 3d transform-domain collaborative filtering," in *European Signal Processing Conf.*, 2007.

[36] C.-K. Liang, T.-H. Lin, B.-Y. Wong, C. Liu, and H. H. Chen, "Programmable aperture photography: multiplexed light field acquisition," in *ACM Trans. Graphics (TOG)*, 2008.

[37] J. Yang, K. Yu, and T. Huang, "Supervised translation-invariant sparse coding," in *IEEE Conf. Computer Vision and Pattern Recognition*, 2010.

[38] F. Heide, W. Heidrich, and G. Wetzstein, "Fast and flexible convolutional sparse coding," in *IEEE Conf. Computer Vision and Pattern Recognition*, 2015.

[39] H. Bristow, A. Eriksson, and S. Lucey, "Fast convolutional sparse coding," in *IEEE Conf. Computer Vision and Pattern Recognition*, 2013.

[40] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *Curves and Surfaces*. Springer, 2012, pp. 711–730.