Dictionary Learning for Scalable Sparse Image Representation with Applications

Bojana Begovic\(^*\), Vladimir Stankovic, Lina Stankovic

Department of Electrical and Electronic Engineering, University of Strathclyde, Glasgow, G1 1XW, UK
\(^*\)Corresponding Author: bojana@begovic.co.uk

Abstract This paper introduces a novel design for the dictionary learning algorithm, intended for scalable sparse representation of high motion video sequences and natural images. The proposed algorithm is built upon the foundation of the K-SVD framework originally designed to learn non-scalable dictionaries for natural images. Proposed design is mainly motivated by the main perception characteristic of the Human Visual System (HVS) mechanism. Specifically, its core structure relies on the exploitation of the high-frequency image components and contrast variations in order to achieve visual scene objects identification at all scalable levels. Proposed design is implemented by introducing a semi-random Morphological Component Analysis (MCA) based initialization of the K-SVD dictionary and the regularization of its atom’s update mechanism. In general, dictionary learning for sparse representations leads to state-of-the-art image restoration results for several different problems in the field of image processing. In experimental section we show that these are equally achievable by accommodating all dictionary elements to tailor the scalable data representation and reconstruction, hence modeling data that admit sparse representation in a novel manner. Performed simulations include scalable sparse recovery for representation of static and dynamic data changing over time (e.g., video) together with application to denoising and compressive sensing.

Keywords Scalable Video Representation, Sparse Coding, Regularization, Contrast Variation, Denoising, Compressive Sensing

1 Introduction

Over the past couple of decades, image processing applications have undergone significant improvements. A recent critical factor in this growth is the sparse coding paradigm introduced firstly by [1], based on the assumption that signals (e.g., natural images) admit a sparse decomposition over a learned representational basis i.e., dictionary. This so-called sparseland model [2, 3, 4] has led to numerous state-of-the-art algorithms for several image processing problems [3] specifically in the context of dictionary \( \mathbf{D} \in \mathbb{R}^{n \times K} \) learning for any image signal class. Commonly, the representation of image \( \mathbf{Y} \in \mathbb{R}^{n \times n} \) is broken down into a set of \( N \) extracted patches \( \{\mathbf{y}_i\}_{i=1}^N \in \mathbb{R}^n \) which are in turn sparsely represented. Typically (but not necessarily) it is assumed that dictionary \( \mathbf{D} \) is overcomplete i.e., the number of its basis vectors (atoms) is greater than the original signal’s dimension \( (K > n) \). Given one of the pursuit algorithms e.g., [5, 6, 7, 8, 9] and a dictionary \( \mathbf{D} \), one can estimate matrix \( \mathbf{X} \) containing sparse approximations \( \{\mathbf{x}_i\}_{i=1}^N \in \mathbb{R}^N \) for each \( \mathbf{y}_i \). Hence, a set of weighted linear combinations of few atoms in \( \mathbf{D} \) satisfactorily approximates each patch \( \mathbf{y}_i \in \mathbf{Y} \) with image denoted as \( \mathbf{Y} \approx \mathbf{D}\mathbf{X} \). The applications of dictionary learning [10, 11] include areas such as classification [12, 13], efficient face recognition [14], inpainting [15], denoising [16, 17], super-resolution [18, 19], Morphological Component Analysis (MCA) [20, 21] and those designed for sparse color image processing [22, 23].

In this paper, we provide a detailed presentation of the scalable and sparse modeling dictionary learning framework which basic outline was originally presented in [24, 25]. Our focus is placed on designing a procedure for learning a dictionary capable of adapting both to a specific dataset and providing its effective scalable reconstruction. Given that current work on scalable data recovery is only based on the the conventional predefined dictionaries such as DCT [26] we find that it is important to offer an alternative one in a form of an adaptive dictionary sparse representation. In addition, to the best of our knowledge existing literature just provides the dictionary learning algorithms such as K-SVD [3, 10] that only assume fine resolution as the representational output. This is not sufficient nor tailored to provide the progressive image recovery over its trained sparse representation. Thus, we take and extend the classical form of the K-SVD where the proposed learning scheme differs from the regular one [3, 10] in the sense that:

- Dictionary is initialized in a controlled, semi-random manner using image’s MCA properties [20, 21];
- A novel learning design is introduced as a regular-
ization of a dictionary training procedure prior to Singular Value Decomposition (SVD) [27, 28];

- It allows more flexibility for adapting the scalable representation to specific data by removing constraints originally imposed on redundant atoms (i.e., mutually coherent or rarely used) in [5];

- A firm spatial frequency distribution is enforced over dictionary atoms as a built-in feature;

- OMP pursuing algorithm is replaced via a simple matrix inversion for sparse coefficient estimation given that proposed dictionary is complete.

Specifically, proposed implementation is carried out by introducing regularization of the K-SVD atoms update stage aiming for scalable sparse image reconstruction which would improve gradually as we take more and more entries per each coefficient $x_i \in X$ to restore \( \{y_i\}_{i=1}^{N} \) patches. In particular, we emphasize the penalization of the low and high spatial frequency components of the image patches and dictionary, imposing the learning model that mimics the main HVS [29, 30] system properties. That is, incorporating HVS’s high sensitivity to contrast light information and to the patterns orientation [31, 32] at high spatial frequencies (originally shown via contrast sensitivity function map [33, 34]). The HVS features are proven to be essential modeling elements for many image processing algorithms [35] and image quality assessment tools [36, 37].

Furthermore, in modern video broadcasting networks, an image or a video source is transmitted to numerous clients with various receiver characteristics. These consumers differ primarily in accessible: (i) channel capacity; (ii) display resolutions; (ii) computing resources. The interesting question is how to support and deliver a controlled quality of the displayed data of a wide range of applications that differ in the users equipment heterogeneity, communication channels and QoS demands? It would be appealing somehow for a video or image signal to be processed in a such manner that would enable its optimal usability by all diverse clients. For example, the limited frequency space shared by mobile video streaming users would be effectively exploited by a generic scalable i.e., progressive data reconstruction such as proposed here. In other words, progressive reconstruction framework would be applied on the source signal prior to its transmission producing its scaled representation form. Once delivered at the client side, depending on its technical specifications, signal would be restored at different quality levels. Thus, signal’s generic scalability is desirable in many applications since it will be able to support heterogeneity in users equipment, QoS demands, and communication channels.

This paper makes following contributions:

1. It tackles for the first time the problem of creating a dictionary tailored to scalable image restoration, offering a novel model for data that admits sparse representation;

2. Enforces specific spatial frequency distribution as a built-in feature over trained dictionary;

3. As a solution to the scalable image restoration problem, this paper provides an extension and upgrade of the K-SVD dictionary learning concept from non-scalable to scalable adaptive image reconstruction by introducing semi-random dictionary initialization based on the MCA activity norm [3] and by regularizing the learning process of dictionary elements overall promoting the HVS perceptual mechanism features;

4. The potential of the proposed method is shown for the adaptive scalable denoising and CS;

Evaluation of the scalable recovery is done using high-motion test video sequences and several natural images, successfully attaining progressive frame-to-frame and image scalable restoration. Experimental results confirm that the proposed scalable scheme outperforms significantly conventional K-SVD at different scalable image recovery levels. Specifically, in terms of application, our focus is placed on applying our proposed scalable scheme to denoising [15] and compressive sensing (CS) [5, 38, 39, 40, 41]. Mainly, in relation to denoising, we tackle processing and computational demands of the [16] given that experimental results in [42] suggest that objective quality improvement of current state-of-the-art image denoising schemes cannot be improved by more than 0.1 [dB]. This conclusion is a result of comparison between the lowest error rates given a simple statistical measure derived from a huge image patch distribution [42] and the empirical errors of state-of-the-art denoising algorithms. Lastly, given the CS results in [43, 44] we test the performance of proposed scalable dictionary learning method in one of the CS sampling scenarios.

2 Problem statement and proposed approach

Adhering closely to the notation used in [10], this section provides the detailed description of the proposed dictionary learning scheme for scalable image reconstruction. We build on the regular K-SVD algorithm [10] by altering its initialization and atom’s update step. In general, we are given a set of $N$ signals i.e., overlapping image patches of size $\sqrt{m} \times \sqrt{n}$ vectorized as $Y = [y_1, \ldots, y_N] \in R^{n \times N}$. The classical configuration of the K-SVD algorithm aims to approximate representation of these signals in a sparse way as weighted linear combinations of a few dictionary elements i.e., the columns of matrix $D = [d_1, \ldots, d_K] \in R^{n \times K}$.

Note however that this conventional approach is not capable of providing scalable image reconstruction that would be based on progressive recovery of each image patch $y_i$. For instance, one can form \{a \mid 1 \leq a \leq \lfloor K/m \rfloor = s\} number of recovery layers for each patch leading to reconstructed image denoted as $L_0$. In general, $m$ can vary and take on different values i.e., $1 \leq m \leq K$ resulting in a number of scalable recovery layers having $m$ as the scaling parameter. This leads to a progressive image restoration provided as a sequence of $L_0$ image layers each generated as a combination of the truncated versions of sparse representation $X$ and dictionary $D$. At the beginning of the progressive
recovery, the base layer $L_1$ is rebuilt out of the first $m$ sparse coefficients entries per patch. That is, for each patch $i$ we take $[x_i[1], x_i[2], \ldots, x_i[m]]$ while remaining entries are set to zero $x_i = 0$ for $m < i \leq K$. These are combined together with the first $m$ corresponding atoms i.e. $[d_1, d_2, \ldots, d_m]$ leading to a compression rate of $m/n$. Afterwords, while reconstructing each subsequent layer $L_a$ $(a > 1)$ additional $m$ coefficients are added. That is, $[x_i[1], x_i[2], \ldots, x_i[m]]$ $(x_i = 0$ for $am < i \leq K)$ and $[d_1, d_2, \ldots, d_m]$ producing compression ratio of $(am)/n$.

The way in which we achieve effective sparse adaptive scalable image reconstruction is by introducing:

1. MCA based semi-random initialization of the dictionary at the very beginning of the training procedure;
2. A regularization scheme over the second K-SVD iterative stage i.e., the dictionary atom’s update which enforces significance of the high frequency components during the regularized atom’s update.

The following terms will be used in the remainder of this paper:

- $Y \in R^{n \times N}$ - matrix with $N$ overlapping image patches $y_i \in R^n$;
- $D_{sc} \in R^{n \times K}$ - proposed scalable dictionary;
- $D \in R^{n \times K}$ - conventional non-scalable dictionary obtained using standard K-SVD [1][5];
- $K$ - the number of dictionary atoms in $D_{sc}$ or $D$;
- $X \in R^{K \times N}$ - sparse matrix with sparse coefficient vectors $x_i \in R^K$.

2.1 Dictionary initialization

In classical K-SVD, prior to any of the two training stages, dictionary $D$ is initialized with $K$ randomly extracted image training patches $y_i$ [10] from the set of total $N$. In contrast, prior to initialization we divide the $N$ training patches in two classes $C_1$ and $C_2$ each containing smooth and texture image content, respectively. As a classification criteria we use the activity measure similar to TV norm originally used within the K-SVD MCA setup [3] and defined as:

$$\text{Activity}(y_i) = \sum_{j=2}^{n} \sum_{k=1}^{m} |y_i[j,k]-y_i[j-1,k]| + \sum_{j=1}^{n} \sum_{k=2}^{m} |y_i[j,k]-y_i[j,k-1]|. \tag{1}$$

Subsequently, Activity is normalized in a way which sets its range from 0 to 1. These values are reflecting the degree of “smoothness” and “texturenness” in each image patch [3]. The higher the Activity the higher the level of the texture will be within the patch. Thus, the classification is performed via simple tresholding using heuristically set value $A$. This value is taken from [3] where it is shown that it provides the best possible classification performance for smooth and texture element separation. Specifically, classifying parameter $A$ indicates classification of patches into two classes $C_1$ or $C_2$. That is:

- $y_i \in C_1$ for $\text{Activity}(y_i) \leq A$;
- $y_i \in C_2$ for $\text{Activity}(y_i) > A$.

Thereafter, the first $K/2$ atoms of the proposed dictionary $D_{sc}$ are initialized randomly choosing $K/2$ image patches from the $C_1$ class, that is, the smooth group. The rest of the $K/2$ atoms are randomly picked from the $C_2$ class i.e., the texture group. In this way, we enforce semi random initialization which directly controls and effects the starting dictionary structure by placing low frequencies (smooth image areas) within its first half of $d_j$ atoms $(1 \leq j \leq K/2)$ and high ones (texture image areas) within the last half $(K/2 < j \leq K)$. In return, this sets a foundation for further design which is organized around applying proposed regularization scheme and subsequently tuning dictionary learning to the main HVS perception characteristic.

2.2 Sparse coding

The first of the two iterative dictionary learning stages (sparse coding) is posed as a constraint optimization problem defined in [10] as:

$$\min_{X} \left\{ \|Y - DX\|_F^2 \right\} \text{ s.t. } \forall i \quad \|x_i\|_0 \leq T_0 \tag{2}$$

given the current estimation of the dictionary $D$ which is kept fixed during this process. The expression $\|x_i\|_0$ accounts for the number of non-zero elements in each vector $x_i$ by means of the $l_0$ pseudo norm where $T_0$ impose the top limit on $\|x_i\|_0$. Signal $y_i \in Y$ $(i = 1, \ldots, N)$, extracted from the original image, is mapped into its sparse representation $x_i$ commonly via OMP [3, 6]. However, given that we train a complete dictionary as proposed, OMP is not needed for the sparse coding step. That is, the exact solution for the scalable dictionary is attained via simple matrix inversion as $x_i = D_{sc}^{-1}y_i$ by maintaining up to $T_0$ largest non-zero coefficient entries. Each of $K$ entries $x_i[j]$ corresponds to one of the atoms $d_j \in D_{sc}$ $(j=1, \ldots, K)$ where nonzero entry $x_i[j] \neq 0$ means that particular atom $d_j$ participates in the sparse representation of the signal $y_i$ [10].

We relax the sparsity constraint, permitting $T_0$ to take a higher value than in [10] where the relation $T_0 < \leak n$ is still maintained. This allows the scalable signal recovery to be established while introducing a $T_0$ value on an empirical basis that still promotes the sparsity prior of the signal.

2.3 Regularized dictionary update stage

Once stage described in 2.2 is completed, we move to the atom $d_j$ update stage. Usually, the new basis atom is estimated by processing a current representation residual $E_j$ [10] constructed to account for the error of all $N$ patches when the atom $d_j$ is removed:

$$\left\| \left( Y - \sum_{k \neq j} d_kx_k^T \right) - d_jx_j^T \right\|_F^2 = \left\| E_j - d_jx_j^T \right\|_F^2 \tag{3}$$

$x_j^T$ represents coefficients from the $k_{th}$ row in $X$ (3) [9] where $x_j^T[i] \neq 0$ denotes that the sparse approximation
introduced regularization over the smooth and texture learned representation. Outcomes show that carefully v
the low and high frequency components of the training of all patches in E
ried out by dividing the current sparse approximations scalable
rons at the moment of perception, primarily identifyingject given the high firing rate of the visual cortex neu-
E
of the
ation. That is why we decide to redefine the structure of testing various weight pairs \[ d_j \]

This separation is plausible due to semi-random initialization described in Sec. 2.1. Proposed design is a result of testing various weight pairs \([v_0, v_1]\) under the constraint \(v_0 + v_1 = 1\) in order to avoid degeneracy of the learned representation. Outcomes show that carefully introduced regularization over the smooth and texture image components is able to yield the appropriate dictionary for the scalable data representation (Sec. 3).

Further, by introducing:

- \( Y_j = Y \Omega \)
- \( D_{sc}^{\text{low}} X_j^{\text{low}} = \left( \sum_{k=1}^{K/2} d_k x_T^k \right) \Omega_j \)
- \( D_{sc}^{\text{high}} X_j^{\text{high}} = \left( \sum_{k=K/2+1}^{K} d_k x_T^k \right) \Omega_j \)

the proposed regularized error matrix (4) can be rearranged as:

\[
E_R^j = \left( Y_j - v_0 D_{sc}^{\text{low}} X_j^{\text{low}} - v_1 D_{sc}^{\text{high}} X_j^{\text{high}} \right)
\tag{5}
\]

where \( Y_j \) represents a subset of the image patches \( y_i \) from \( Y \) with indices given in \( \omega_j \). Superscripts low and high denote smooth and texture frequency content associated with the weight pair \([v_0, v_1]\) which regularizes contribution of their residual components to the \( E_R \).

Consequently, this separation controls the type of the information used for the \( d_j \) atom's update.

For all atoms \( j = 1, ..., K \), the proposed update stage is summarized as:

1. **STEP 1** - Allocate corresponding image patches which current sparse approximation given as a linear superposition \( D_{sc} x \) includes atom \( d_j \) as it is done in [10], map them accordingly with \( \omega_j \) and denote as a subset of patches \( Y_j \), that is a subset of sparse coefficients \( X_j \).

2. **STEP 2** - In contrast to [10], split each current sparse approximation element \( x_i \in X_j \) \( (i \in \omega_j) \), associated with atom \( d_j \), in two parts using binary vectors \( T^{\text{low}}, T^{\text{high}} \in R^K \) as:

\[
\begin{align*}
x_i^{\text{low}} &= x_i^{T^{\text{low}}} \\
x_i^{\text{high}} &= x_i^{T^{\text{high}}}
\end{align*}
\]

where \( T^{\text{low}}, T^{\text{high}} \in R^K \) are binary vectors that cancel any \( x_i \) [l] element for \( l > \frac{K}{2} \) (associated with the dictionary elements initialized with class \( C_1 \)) and \( l < \frac{K}{2} \) (associated with the dictionary elements initialized with class \( C_2 \)) as follows:

\[
\begin{align*}
T^{\text{low}} [l] &= \begin{cases} 1 & \text{if } l \leq \frac{K}{2} \\ 0 & \text{if } l > \frac{K}{2} \end{cases} \\
T^{\text{high}} [l] &= \begin{cases} 0 & \text{if } l \leq \frac{K}{2} \\ 1 & \text{if } l > \frac{K}{2} \end{cases}
\end{align*}
\]

In this way the smooth and texture patch content are extracted finally as \( D_{sc}^{\text{low}} X_j^{\text{low}} \) and \( D_{sc}^{\text{high}} X_j^{\text{high}} \), respectively.

3. **STEP 3** - After decomposing sparse representation of \( Y_j \) accordingly form newly proposed representation residual error term \( E_j^R \) associated with atom \( d_j \) as (5);

4. **STEP 4** - Perform rank-one approximation of \( E_j^R \) via SVD and set the eigenvector corresponding to the largest eigenvalue as new \( d_j \) and the \( |\omega_j| \) largest eigenvalues as the new non-zero entries for the \( x_T^j \) (as in [10]);

5. **STEP 5** - Keep redundant atoms (unlike [10]): mutually coherent and rarely used ones;
Proposed regularization plays an important role given that weights \( v_0 \) and \( v_1 \) control which spatial frequency content will be joined to the \( E^R \). Consequently, the SVD decomposition (STEP 4) generates atoms of the scalable dictionary \( D_{sc} \) based on the information contained within \( E^R \). By keeping more of the original high frequency info \((v_1 < 0.5)\) and suppressing the lower one \((v_0 > 0.5)\) the algorithm regularizes the learning process which effectively generates dictionary \( D_{sc} \) suitable for scalable representation. This enables recovery of the basic image objects shapes from the base layer \( L_1 \) resulting in a learning procedure which is tailored to the characteristics of HVS.

### 2.4 Denoising and scalable dictionary scheme

Prior to presenting the scalable denoising process, we inspect the way in which noise is removed during the classical K-SVD dictionary training. Commonly, noise is iteratively discarded throughout two stages:

1. While performing sparse coding, OMP stops when the current approximated sparse solution reaches the sphere of radius \( \sqrt{n} \sigma \) in the patches space. This radius constrains the acceptable level of the recovered noise strength i.e., \( |e|^2 \leq Cn\sigma^2 \). Going bellow this boundary would result in direct noise reconstruction where C is a heuristically set constant and \( \sigma \) stands for the noise standard deviation;

2. During the dictionary’s atom’s update, noise is removed via SVD decomposition that estimates new “average” direction for each atom least influenced by the distortion.

The conventional K-VD denoising energy minimization problem [16, 17] is given as:

\[
\{ \hat{x}_i, \hat{D}, \hat{y}_i \} = \text{arg min}_{x_i, D, y_i} \lambda \|y_i - y_i^{noisy}\|_2^2 + \sum_i \mu_i \|x_i\|_0 + \sum_i \|Dx_i - y_i\|_2^2 \tag{6}
\]

We simplify this complex minimization task by relaxing the regularization process with the introduction of the proposed scalable dictionary \( D_{sc} \) as follows:

\[
\min_{D_{sc}, y_i} \lambda \|y_i - y_i^{noisy}\|_2^2 + \sum_i \|D_{sc}x_i - y_i\|_2^2 \tag{7}
\]

In (7) we decide to discard the sparse coding phase while merely performing noise removal during the scalable dictionary \( D_{sc} \) update. Our detailed study of denoising scheme in [16, 17] suggests that the initial sparseness level i.e., the average number of the non-zero coefficients nearly stays fixed during the dictionary training in the classical K-SVD setup. That is, the one established after the first OMP sparse coding over the initialized dictionary [16]. Furthermore, we impose assumption that the noise less distorts texture than smooth image components due to the high-frequency nature of the texture information. Justification for this is provided in Sec. 4.3 where we illustrate how various level of noise distort smooth and texture image blocks based on their standard deviation estimated before and after noise is introduced.

Thus, we promote the idea that, after the initial matrix inversion \( X = D_{sc} Y \) (substitute for OMP), we could neglect subsequent ones during the dictionary learning while still obtaining a satisfactory denoising results given that texture information prevails for the modified dictionary update. For this setup, the coefficient entries \( x_{ij} \) are only updated during the SVD decomposition employed for the atom’s update. Hence, the introduced modification is expected to result in a considerably shorter computational processing time while achieving comparable quality as obtained with the non-scalable K-SVD denoising scheme.

### 2.5 Computational Complexity

The proposed design does not incur the cost of the original dictionary learning in [3, 10] in case of training strictly representative dictionary over the noise free image. Given that there are no additional transforms employed but just linear separation of the low and high frequencies components via semi-random initialization and introduced error matrix regularization (as shown in Sec. 2.3) the computational complexity remains of the same order as that of the conventional non-scalable K-SVD. That is, the number of operations per pixel is still \( O(nT_0I) \) where \( I \) stands for the number of iterations. By setting the number of atoms \( K = n \) and replacing OMP via simple matrix inversion, we manage to even decrease the processing demands while achieving good signal recovery (typically in [3, 10] \( K \) is equal to \( 2n, 3n \) or \( 4n \)). This is in particular transparent in relation to when applying proposed scheme to denoising given that sparse coding stage is removed. More details on processing time necessary for denoising are shown in Sec. 3.2.

### 3 Simulation results

The performance of the proposed scalable K-SVD method is evaluated in the set of experiments applied to:

- Standard CIF high motion video test sequences “Stephan” and “Tempete” at resolution 352 \( \times \) 288 and a frame rate of 30 Hz;
- Several natural images e.g., “Boat” and “Peppers”, size 512 \( \times \) 512.

Variables and parameters for all simulations are summarized in Tab. 1 together with their values and roles. Prior to processing, every frame is broken down into \( N = 96,945 \) or every natural image into \( N = 255,025 \) overlapping patches size 8 \( \times \) 8 pixels. Thus, the vectorized dimension of the signals used for the scalable dictionary learning algorithm is \( n = 64 \) pixels, while both dictionaries \( D_{sc} \) and \( D \) contain \( K = n \) atoms with redundancy factor \( r = 1 \). Sparsity level \( T_0 \) is set to 10 both for training and reconstruction phase. This provides the best processing effectiveness (in terms of PSNR values) for the proposed scalable learning design after testing the wide range of sparsity levels e.g., \( T_0 = [4,5,\ldots,28] \).
The number of progressively recovered layers $L_a$ is defined with scaling parameter $m = 4$ as $[K/m] = s = 16$ for every layer of scalable patch recovery and therefore image with $a = 1, \ldots, 16$.

Starting from the first scalable version of the processed image i.e., base layer $L_1$, the reconstruction is carried out using only first $m = 4$ entries (i.e., $6.25\%$) per each sparse coefficient $\mathbf{x}_i$. This first level of truncated coefficients from $\mathbf{X}$ is denoted with $\mathbf{X}_1 \in \mathbb{R}^{(4) \times N}$. Along this, we employ a truncated version of trained dictionary $\mathbf{D}_{sc}^1$: $\mathbf{D}_{sc}^1$ with only first $m = 4$ atoms i.e., $[\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3, \mathbf{d}_4]$. The remaining recovery levels are progressively enhanced by adding four (i.e., $m$) additional entries in each representational vector and four (i.e., $m$) additional atoms. In general, (for any $m$ value) the progressive recovery of the each image patch $\mathbf{y}_i$ for new scalable layer $L_a$ starts by taking all $m(a - 1)$ entries from the associated sparse coefficient $\mathbf{x}_i$ previously employed for the estimation of the $\mathbf{y}_i$ scalable version at the level $L_{a-1}$. This continues by adding subsequent $m$ values from the sparse coefficients $\mathbf{x}_i$ that are indexed as:

- $\mathbf{x}[m(a - 1) + 1]$;
- $\mathbf{x}[m(a - 1) + 2]$;
- $\ldots$
- $\mathbf{x}[m(a - 1) + m]$;

with reconstruction for scalable patch at $L_a$ given as $\mathbf{x}^m \mathbf{x}^m$ with $\mathbf{x}^m \in \mathbf{X}^m$. The end result is that each recovered patch at the new layer $L_a$ will contain the first $m(a - 1)$ reconstructed elements as patches in $L_{a-1}$ and newly estimated $m$. For shown case, this is done until the final $L_{16}$ restoration level is attained using full sparse representation $\mathbf{X}_{16} = \mathbf{X}$ and all atoms in dictionary $\mathbf{D}_{sc}^{16} = \mathbf{D}_{sc}$. That is, the recovery scheme for each image layer is given as:

- $L_1 = \mathbf{D}_{sc}^1 \mathbf{X}_1$: 4 atoms and entries per sparse coefficient;
- $L_2 = \mathbf{D}_{sc}^2 \mathbf{X}_2$: 8 atoms and entries per sparse coefficient;
- $\ldots$;
- $L_{15} = \mathbf{D}_{sc}^{15} \mathbf{X}_{15}$: 60 atoms and entries per sparse coefficient;
- $L_{16} = \mathbf{D}_{sc}^{16} \mathbf{X}_{16}$: 64 atoms and entries per sparse coefficient.

Unlike the classical, non-scalable sparse dictionary learning where practice is to train an overcomplete dictionary ($K >> n, r > 1$), we promote training of a complete one. One of the main reasons for this arises from our observation that whether we train a complete or overcomplete basis $\mathbf{D}_{sc}$, the achieved restoration quality for scalable image representation is highly comparable. Tab. 2 and Tab. 3 show the averaged comparison at every scalable recovery level $L_a$ given the complete and overcomplete scalable scheme for video sequence “Stephan” and image “Peppers”, respectively. The number of atoms for the overcomplete $\mathbf{D}_{sc}$ dictionary is $K = 128$ ($r = 2$) thus having greater number of the recovery levels $[K/m] = s = 32$ than with the complete scheme given the same scaling factor $m = 4$.

On average, when we take into account all testing results, the difference of the highest recovered layers i.e., $L_{16}$ goes around 0.15[dB] (for frame size $352 \times 288$) and 0.66 [dB] (for image size $512 \times 512$) in favor of the overcomplete $\mathbf{D}_{sc}$ dictionary. We can see a comparison of every two recovery levels of the overcomplete $\mathbf{D}_{sc}$ dictionary with one of the recovery level of the complete $\mathbf{D}_{sc}$ in Tab. 2 and Tab. 3 (e.g., $L_5$ and $L_{26}$ with $L_{13}$). Conclusion follows that the scalable performance of the complete $\mathbf{D}_{sc}$ overruns the overcomplete $\mathbf{D}_{sc}$ at all recovery levels (bold values) except for the final $L_{16}$ that is $L_{52}$ for “Stephan” and almost all levels for “Peppers”. Having this in mind together with the fact that lesser number of trained atoms:

- Minimizes the amount of information necessary for training and signal’s recovery;
- Lowers computational complexity;

we chose $K = 64$. Prior to defining the proposed scalable scheme, we performed exhausting simulations in order to evaluate performance given the various $P_i$ regularization parameters pairs $[v_0, v_1]$ listed as:

1. $P_1$: $[v_0, v_1] = [0, 1]$;
2. $P_2$: $[v_0, v_1] = [0.1, 0.9]$;
3. $P_3$: $[v_0, v_1] = [0.3, 0.7]$;
4. $P_4$: $[v_0, v_1] = [0.5, 0.5]$;
5. $P_5$: $[v_0, v_1] = [0.7, 0.3]$;
6. $P_6$: $[v_0, v_1] = [0.9, 0.1]$;
7. $P_7$: $[v_0, v_1] = [1, 0]$;

searching for the one which provides the most effective dictionary $\mathbf{D}_{sc}$ for scalable image restoration. Fig. 1 and Fig. 2 present the averaged PSNR and SSIM estimates at every recovery layer $L_a$ of scalable restoration given the high motion video sequence “Stephan” and 10 averaged iterations of the natural image “Peppers” with $K = 64$ number of atoms. As we can see, out of seven $P_i$ regularization ($1 \leq i \leq 7$) scenarios (Fig. 1 and Fig. 2), the $P_2$ results with the dictionary that is most effectively tailored to the scalable sparse image representation given that, overall, results with the highest PSNR and SSIM restoration values. Similar results are achieved for video sequence “Tempete” and several other natural images such as “Boat”. For each of the testing, Activity measure is set to the $A = 0.27$ as in [3]. In addition, we provide results in Fig. 3 (PSNR) and Fig. 4 (SSIM) for all $P_i$ regularization variations when training overcomplete $\mathbf{D}_{sc}$ dictionary again with $K = 128$ atoms for the $K/m = 32$ number of $L_a$ recovery levels. Again, $P_7$ achieves most optimal scalable recovery performance.

To reiterated, the $v_0$ is associated with the $\mathbf{D}_{sc}^{low}$ elements, which capture spatial low-frequencies. These atoms represent a compositional structure of patches extracted from large, smooth, low-variance areas, lacking in harsh edges i.e., the tennis field in the “Stephan” sequence, or the sky background in the “Tempete” sequence. On the other hand, $v_1$ weights the contribution
of the $D_{sc}^{\text{high}}$ atoms that contain higher spatial frequencies, that is, the areas of high detail with many contrastings edges such as the audience in “Stephan” or the flower object in “Tempete”. By looking at (5) in Sec. 2.3, with $P_I$ parametrization ($[v_0, v_1] = [1, 0]$), we can conclude that the regularization process will in each iteration:

- Discard texture sparse approximation $D_{sc}^{\text{high}} x_j^{\text{high}}$ given the $v_1 = 0$;
- Keep the smooth part $D_{sc}^{\text{low}} x_j^{\text{low}}$ with $v_0 = 1$.

This will determine the final content of the regularized error matrix $E_{x_j}^R$ where the texture patches (Activity($y_i$) > 0.27) are dominant information being directly included into the error synthesis rather than being just a part of the representational residual as the smooth one (Activity($y_i$) ≤ 0.27). Furthermore, Sec. 4 provides a detailed discussion on effective structural differences between trained dictionaries and how $D_{sc}$ is better tailored to the HVS perception system than the non-scalable, conventional dictionary $D$. Some of these dictionaries are shown in Fig. 5a and Fig. 5b illustrating examples of scalable (“SC K-SVD”) and non-scalable (“NSC K-SVD”) dictionary trained over the first frame of the “Stephan” sequence. Given the first frame for either of sequences i.e., the training frame, the intro-

| Parameter | Definition | Role |
|-----------|------------|------|
| $N = 96,945$ | Number of image patches | Limits the size of the training set for frames size 122 × 288 |
| $N = 255,025$ | Number of image patches | Limits the size of the training set for images size 512 × 512 |
| $m = 64$ | Constant integer | Dimension of image patch vector and each atom |
| $K = 64$ | Number of dictionary atoms | Limits the size of the representational basis |
| $K/n = r = 1$ | Redundancy factor | Defines overcompleteness of the dictionary |
| $v_0 = 1$ | 1° regularization parameter | Weights smooth patch sparse presentation for the atom’s update |
| $v_1 = 0$ | 2° regularization parameter | Weights texture patch sparse presentation for the atom’s update |
| $\delta = 0.27$ | Activity measure | Classification threshold for smooth and texture image patches $y_i$ |
| $z_{sc} = 10$ | Sparsity level | Limits the number of non zero entries per sparse coefficient $x_i$ |
| $l \in [1, \ldots, 64]$ | Integer index | Defines the entry for a sparse coefficient $x_i$ |
| $m = 4$ | Scaling parameter | Defines number of scalable layers |
| $[K/m] = a = 16$ | General scalability level | Total number of the scalable layers |
| $L_s = 9$ | US scalability level | Limits number of the scalable recovery layers for CS setup |
| $s_1 < s_2, \ldots, < s_L$ | Progressive CS samples | Limit number of samples per each CS scalable recovery layer |
| $S = s_L = 50$ | Maximal CS sample | Limits total number of the CS samples |

Table 1. Table of parameters

Table 2. Averaged PSNR quality assessment for scalable restoration given the “Stephan” video sequence for two sizes dictionary, $K = 64$ and $K = 128$.

| $K = 128$ Stephan [dB] | $K = 64$ Stephan [dB] |
|------------------------|------------------------|
| $L_{32}$               | 30.62                  |
| $L_{31}$               | 29.93                  |
| $L_{30}$               | 27.92                  |
| $L_{29}$               | 26.41                  |
| $L_{28}$               | 26.16                  |
| $L_{27}$               | 25.86                  |
| $L_{26}$               | 25.25                  |
| $L_{25}$               | 24.97                  |
| $L_{24}$               | 24.67                  |
| $L_{23}$               | 24.13                  |
| $L_{22}$               | 23.78                  |
| $L_{21}$               | 23.42                  |
| $L_{20}$               | 22.91                  |
| $L_{19}$               | 22.39                  |
| $L_{18}$               | 22.22                  |
| $L_{17}$               | 21.94                  |
| $L_{16}$               | 21.50                  |
| $L_{15}$               | 20.36                  |
| $L_{14}$               | 17.32                  |
| $L_{13}$               | 16.14                  |
| $L_{12}$               | 15.03                  |
| $L_{11}$               | 13.79                  |
| $L_{10}$               | 10.66                  |
| $L_{9}$                | 9.51                   |
| $L_{8}$                | 8.92                   |
| $L_{7}$                | 8.55                   |
| $L_{6}$                | 8.06                   |
| $L_{5}$                | 7.36                   |
| $L_{4}$                | 5.90                   |
| $L_{3}$                | 5.52                   |
| $L_{2}$                | 5.16                   |
| $L_{1}$                | 5.01                   |

Table 3. Averaged PSNR quality assessment for scalable restoration given the “Peppers” natural image for two sizes dictionary, $K = 64$ and $K = 128$.

| $K = 128$ Peppers [dB] | $K = 64$ Peppers [dB] |
|------------------------|------------------------|
| $L_{32}$               | 35.78                  |
| $L_{31}$               | 34.59                  |
| $L_{30}$               | 34.45                  |
| $L_{29}$               | 33.52                  |
| $L_{28}$               | 31.57                  |
| $L_{27}$               | 30.62                  |
| $L_{26}$               | 30.46                  |
| $L_{25}$               | 30.10                  |
| $L_{24}$               | 30.06                  |
| $L_{23}$               | 29.96                  |
| $L_{22}$               | 29.66                  |
| $L_{21}$               | 29.24                  |
| $L_{20}$               | 29.01                  |
| $L_{19}$               | 28.70                  |
| $L_{18}$               | 28.51                  |
| $L_{17}$               | 28.41                  |
| $L_{16}$               | 28.25                  |
| $L_{15}$               | 24.20                  |
| $L_{14}$               | 22.11                  |
| $L_{13}$               | 18.09                  |
| $L_{12}$               | 16.08                  |
| $L_{11}$               | 15.06                  |
| $L_{10}$               | 14.07                  |
| $L_{9}$                | 11.33                  |
| $L_{8}$                | 10.99                  |
| $L_{7}$                | 9.25                   |
| $L_{6}$                | 8.97                   |
| $L_{5}$                | 7.97                   |
| $L_{4}$                | 7.68                   |
| $L_{3}$                | 7.40                   |
| $L_{2}$                | 6.12                   |
| $L_{1}$                | 6.35                   |
duced training scheme is carried out only once generating the Dsc dictionary. Subsequently, while reconstructing each incoming frame we use this single Dsc, thus, not training any new dictionary. This approach considerably reduces the computational complexity of the scalable sparse video representation, since training is done only once instead for each incoming frame. This is immensely important in the context of real-time scalable image/video applications development. It is necessary to mention that in general, when the video scene undergoes significant changes with respect to the training frame, a new training frame should be inserted. This is necessary in order to accommodate for the difference in the compositional structure of previous frames and newly changed one.

3.1 Scalability Performance

The comparison of the restoration quality is done for the proposed regularized scalable “SC K-SVD” and conventional non-scalable “NSC K-SVD” algorithm. Fig. 6a and Fig. 6b illustrate the PSNR estimates for video sequences “Stephan” and “Tempete” respectively. Shown results are averaged over all frames given each of 16 recovery L1 layers. Minor exception is the “Stephan” sequence which frames are divided into two groups: [1, 270] and [271, 300]. This frame separation is carried out in order to demonstrate the variation in the quality of the restored image, when a new object is introduced in the frame 271. We would expect certain degradation in the restoration quality given that Dsc is trained over the training frame which does not contain a newly introduced visual object.

Depicted results clearly demonstrate that the proposed regularized scheme considerably outperforms the standard [10] over all recovery levels L1, where average gain of 11.32 [dB] (“Stephan”, first 270 frames) and 8 [dB] (“Tempete”, all frames) is achieved. This proves superiority of the proposed work in terms of scalable recovery. Once a new image object appears (e.g., the tennis net in the “Stephan” sequence) a noticeable drop in the scalable recovery quality can be noticed in Fig. 6a for the second frame group [271, 300]. Specifically, on average “SC K-SVD” PSNR declines for 1.84 [dB] while still outperforming the “NSC K-SVD” for 9.43 [dB]. Over-all, only in the case when all the information on the sparse coefficients is available (X16 ∈ R(64)×N), the regular K-SVD algorithm has a slight advantage over the proposed scheme. Besides the standard objective quality assessment i.e., PSNR, we consider an alternative quality measure, so-called Structural Similarity Index (SSIM) [37]. It is designed to quantify the degree to
which image structural information is degraded by calculating a quality index ranging from 0 (denoting highest distortion) up to 1 (denoting no distortion). This measure is specially appealing for the evaluation of the proposed scalable image restoration framework due to the fact that the SSIM is based on the discussed HVS characteristics. Specifically, it takes into account local pixels distortions of the luminance and contrast information. The higher the SSIM index value gets, the more successful retrieval of the HVS perception information at each scalable layer $L_a$ will be. This results in a better visual information thus providing progressive image recovery of better quality. Therefore, SSIM index values shown in Fig. 7 quantify the degree of the degradation of structural information in a frame at each scalable reconstruction level $L_a$. Once again, these estimates are averaged over all frames for both testing video sequences. As in the case of PSNR, in Fig. 7a, for the “Stephan” sequence, we can see that the proposed scalable method surpasses in general the non-scalable for 0.37 (first frame
Figure 6. Frame-average PSNR of the scalable reconstructed video test sequences (“Stephan” and “Tempete”) given for each layer $L_a$ of the scalable video reconstruction using the scalable and non-scalable K-SVD algorithm.

(a) “Stephan” video sequence

(b) “Tempete” video sequence

Figure 7. Frame-average SSIM index of the scalable reconstructed video test sequences (“Stephan” and “Tempete”) given for each layer $L_a$ of the scalable video reconstruction using the scalable non-scalable K-SVD algorithm.

(a) “Stephan” video sequence

(b) “Tempete” video sequence

Visualization of the results is provided in Fig. 8, Fig. 9 for “Stephan” and in Fig. 10, Fig. 11 for “Tempete” sequence in order to emphasize the subjective perceptual quality. These figures illustrate the scalable reconstruction outcome at every recovery level $L_a$ for the so-called training frame (Fig. 8 and Fig. 10). The last frames for both video sequences are shown in Fig. 9 and Fig. 11. Here one can observe the visual variations in the restoration quality when the new object containing the high-frequency content structure is introduced (i.e., the tennis net in Fig. 9) or the more spatial low-frequencies are added i.e., the background in “Tempete” in Fig. 11. All scalable restorations are performed over the single trained dictionary given the first training frame. From these figures one can notice that the proposed scalable scheme is able to recover the frame at a recovery level $L_3$ ($D_{3sc} \in \mathbb{R}^{64 \times 12}$ and $X_3 \in \mathbb{R}^{12 \times N}$) whereas [1] fails to show any scalable characteristics overall up to $L_{15}$ ($D_{15sc} \in \mathbb{R}^{64 \times 60}$ and $X_{15} \in \mathbb{R}^{60 \times N}$) for “Stephan” and $L_8$ ($D_{8sc} \in \mathbb{R}^{64 \times 32}$ and $X_8 \in \mathbb{R}^{32 \times N}$) for “Tempete”. It should be said that the “NSC K-SVD” does show slight scalability with the “Tempete” sequence. However, this is still far from the performance of the proposed method that keeps its reconstruction efficiency consistent for quite different video sequences, hence showing its processing stability.

3.2 Application to image processing 1: denoising

This experimental section justifies and validates advantages of the proposed scheme for the denoising application. We compare introduced scalable (denoted as “SC”) over the non-scalable (denoted as “NSC”) together with the overcomplete classical K-SVD dictionary “Org”. For the aforementioned algorithm setups, we discuss objective quality assessment and time processing complexity. However, unlike in Sec. 3.1 where dictionary training is done only once over the first non distorted frame in the video sequence, for denoising, the dictio-
Figure 8. Visual assessment of the scalable reconstruction using the scalable and non-scalable K-SVD at every recovery level $L_a$.

Figure 9. Visual assessment of the scalable reconstruction using the scalable and non-scalable K-SVD at every recovery level $L_a$.

Figure 10. Visual assessment of the scalable reconstruction using the scalable and non-scalable K-SVD at every recovery level $L_a$. 
Dictionary Learning for Scalable Sparse Image Representation with Applications

Figure 11. Visual assessment of the scalable reconstruction using the scalable and non-scalable K-SVD at every recovery level $L_a$.

In addition, we performed testing for the scenario where the sparse coding stage is also removed from the classical non-scalable K-SVD scheme in order to further validate the practicality of the proposed scalable design. After simulations final estimates show that there is a drop of 2 [dB] for “NSC” without sparse coding stage when compared to the best denoising results of the “SC”. Hence, newly introduced regularization scheme is efficient when it comes down to noise removal given that we only keep atom’s update out of two iterative stages for dictionary learning over the corrupted image. The greatest benefit of the scalable denoising is direct reduction of both, computational complexity and processing time where Tab. 8 shows the total denoising run times in seconds for two image sizes:

1. 352x288 - size of the video sequences frames;
2. 512x512 - size of the conventional images.

Illustrated times are outcomes of processing on the Dell operating system with 64 bit Intel core, 8 GB RAM memory and 2.40 GHz processor. The number of iterations for the provided results is fixed and set to sixteen. Based on the averaged run times we can see reduction in:

1. approximately 6.5 times for data of size 352x288 when comparing “SC” vs “Org”;
2. approximately 7.3 times for data of size 512x512 when comparing “SC” vs “Org”;
3. approximately 10.8 times for data of size 352x288 when comparing “SC” vs “NSC”;
4. approximately 11 times for data of size 512x512 when comparing “SC” vs “NSC”;

provided that we achieve still highly comparable (lower levels of noise) or better results (higher levels of noise). The forth column of the Tab. 8 illustrates the time for the error matrix formation per each iteration. These numbers are aiming to show that introduced modification of the atom’s update in the form of a new error matrix scheme influences processing complexity on a minor scale by being increased on average for two seconds.

Finally, Fig. 12, Fig. 13, Fig. 14 and Fig. 15 illustrate visual preview for all discussed data at the recovery level $L_{16}$ after the noise $\sigma = 40$ is removed given the scalable, non-scalable complete or overcomplete K-SVD scheme. After additional subjective quality assessment we can conclude that provided results are highly comparable where, as emphasized, proposed method “SC” put considerably less computational demands than both classical K-SVD setups.
Table 4. PSNR quality assessment for scalable denoising via the scalable and non-scalable K-SVD dictionary, “Stephan” sequence.

| # | PSNR [dB] | Alg. | Scaler recovery levels L | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Org. | NCSC | SC |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 20 | 22.14 | 19.05 | 20.68 | 25.73 | 27.27 | 27.61 | 26.11 | 26.06 | 25.95 | 25.75 | 22.97 | 25.52 | 21.80 | GS |
| 40 | 16.13 | 16.09 | 15.13 | 14.30 | 13.28 | 12.38 | 11.47 | 11.56 | 11.65 | 11.74 | 15.22 | 16.02 | 17.13 | GS |
| 60 | 12.56 | 10.05 | 20.68 | 25.73 | 27.27 | 27.61 | 26.11 | 26.06 | 25.95 | 25.75 | 22.97 | 25.52 | 21.80 | GS |
| 80 | 10.05 | 16.09 | 15.13 | 14.30 | 13.28 | 12.38 | 11.47 | 11.56 | 11.65 | 11.74 | 15.22 | 16.02 | 17.13 | GS |
| 100 | 8.13 | 10.05 | 15.13 | 14.30 | 13.28 | 12.38 | 11.47 | 11.56 | 11.65 | 11.74 | 15.22 | 16.02 | 17.13 | GS |

Table 5. PSNR quality assessment for scalable denoising via the scalable and non-scalable K-SVD dictionary, “Tempete” sequence.

| # | PSNR [dB] | Alg. | Scaler recovery levels L | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Org. | NCSC | SC |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 20 | 22.12 | 19.05 | 20.68 | 25.73 | 27.27 | 27.61 | 26.11 | 26.06 | 25.95 | 25.75 | 22.97 | 25.52 | 21.80 | GS |
| 40 | 16.09 | 16.09 | 15.13 | 14.30 | 13.28 | 12.38 | 11.47 | 11.56 | 11.65 | 11.74 | 15.22 | 16.02 | 17.13 | GS |
| 60 | 12.55 | 10.05 | 20.68 | 25.73 | 27.27 | 27.61 | 26.11 | 26.06 | 25.95 | 25.75 | 22.97 | 25.52 | 21.80 | GS |
| 80 | 10.07 | 16.09 | 15.13 | 14.30 | 13.28 | 12.38 | 11.47 | 11.56 | 11.65 | 11.74 | 15.22 | 16.02 | 17.13 | GS |
| 100 | 8.14 | 16.09 | 15.13 | 14.30 | 13.28 | 12.38 | 11.47 | 11.56 | 11.65 | 11.74 | 15.22 | 16.02 | 17.13 | GS |

Table 6. PSNR quality assessment for scalable denoising via the scalable and non-scalable K-SVD dictionary, “Boat” image.

| # | PSNR [dB] | Alg. | Scaler recovery levels L | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Org. | NCSC | SC |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 20 | 22.12 | 19.05 | 20.68 | 25.73 | 27.27 | 27.61 | 26.11 | 26.06 | 25.95 | 25.75 | 22.97 | 25.52 | 21.80 | GS |
| 40 | 16.09 | 16.09 | 15.13 | 14.30 | 13.28 | 12.38 | 11.47 | 11.56 | 11.65 | 11.74 | 15.22 | 16.02 | 17.13 | GS |
| 60 | 12.55 | 10.05 | 20.68 | 25.73 | 27.27 | 27.61 | 26.11 | 26.06 | 25.95 | 25.75 | 22.97 | 25.52 | 21.80 | GS |
| 80 | 10.05 | 16.09 | 15.13 | 14.30 | 13.28 | 12.38 | 11.47 | 11.56 | 11.65 | 11.74 | 15.22 | 16.02 | 17.13 | GS |
| 100 | 8.15 | 16.09 | 15.13 | 14.30 | 13.28 | 12.38 | 11.47 | 11.56 | 11.65 | 11.74 | 15.22 | 16.02 | 17.13 | GS |

Table 7. PSNR quality assessment for scalable denoising via the scalable and non-scalable K-SVD dictionary, “Peppers” image.

| # | PSNR [dB] | Alg. | Scaler recovery levels L | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Org. | NCSC | SC |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 20 | 22.12 | 19.05 | 20.68 | 25.73 | 27.27 | 27.61 | 26.11 | 26.06 | 25.95 | 25.75 | 22.97 | 25.52 | 21.80 | GS |
| 40 | 16.09 | 16.09 | 15.13 | 14.30 | 13.28 | 12.38 | 11.47 | 11.56 | 11.65 | 11.74 | 15.22 | 16.02 | 17.13 | GS |
| 60 | 12.55 | 10.05 | 20.68 | 25.73 | 27.27 | 27.61 | 26.11 | 26.06 | 25.95 | 25.75 | 22.97 | 25.52 | 21.80 | GS |
| 80 | 10.05 | 16.09 | 15.13 | 14.30 | 13.28 | 12.38 | 11.47 | 11.56 | 11.65 | 11.74 | 15.22 | 16.02 | 17.13 | GS |
| 100 | 8.12 | 16.09 | 15.13 | 14.30 | 13.28 | 12.38 | 11.47 | 11.56 | 11.65 | 11.74 | 15.22 | 16.02 | 17.13 | GS |

Table 8. Comparison of the processing time given tree denoising schemes

| Image size | Alg. | Total denoising run time [s] | Error formation time [s] |
|---|---|---|---|
| 352x288 | Org. | 2897.6 | 1.26 |
| NCSC | 3215.1 | 2.01 |
| SC | 278.4 | 3.42 |
| 512x512 | Org. | 7505.5 | 1.59 |
| NCSC | 7337.3 | 1.58 |
| SC | 999.5 | 3.23 |

3.3 Application to image processing 2: compressive sensing

Following closely the experimental layout suggested in [26], we investigate the effectiveness and the performance of the proposed scalable CS video acquisition scheme when combined with a learned basis rather than a predefined one. In particular, the proposed framework aims for the frame-by-frame progressive CS recov-
ery while analysing the implications of the sub-Nyquist CS paradigm in both the scalable and adaptive representational domain. Likewise in previous experimental sections and as in [26], the image is processed block by block. Mainly, we take into consideration two cases of the CS scalable recovery:

- With the proposed scalable K-SVD dictionary tailored to this task;
- With the conventional non-scalable K-SVD dictionary.

Rather than taking the full number of measurements [5, 45, 46] over every incoming frame, CS sampling is carried out in incremental steps. Note that this is applicable only for the CS scalable sensing scenario. Given the sufficient number of progressive measurements per patch as \( s_1, s_2, \ldots, s_L (s_i < n) \) we are able to recover the frame gradually after incrementally retrieving entries of sparse vector coefficients in \( X_i \) via OMP. Furthermore, each incremental number of samples \( s_i \) satisfies the fundamental result of the CS theory [2] that imposes the limit on the necessary number of measurements for satisfactory signal reconstruction.

Unlike the conventional CS for our testing we apply specially structured sampling matrix \( \Phi \). This aims to achieve efficient scalable acquisition of samples over each image layer commonly denoted as \( Y_{CS} = \Phi y = \Phi D_{\alpha}x \). Implementation is carried out via the systematic non-adaptive approach as in [26] that generates the structural sampling matrix \( \Phi \) optimally suited for the scal-

---

**Figure 12.** Visual assessment for denoising via the scalable, non-scalable complete and over complete K-SVD at \( L_{16} \) recovery level of the first training frame of the “Stephan” video sequence, \( \sigma = 40 \)

**Figure 13.** Visual assessment for denoising via the scalable, non-scalable complete and over complete K-SVD at \( L_{16} \) recovery level of the first training frame of the “Tempete” video sequence, \( \sigma = 40 \)

**Figure 14.** Visual assessment for denoising via the scalable, non-scalable complete and over complete K-SVD at \( L_{16} \) recovery level given the image “Boat”, \( \sigma = 40 \)
able task at hand. For each recovery step (as in [26]) we scale sampling matrix size-wise into its truncated versions as $\Phi_i \in R^{s_i \times n}$. Once the sampling is done we attain a group of samples each denoted as $y_{CS}^i$. The sampling is structured in a way that the basic level is collected via $\Phi_1$ that contains binary entries generated from the Gaussian distribution. Remaining measurements are sampled via Bernoulli binary distributed entries of $\Phi_i$ consecutively added up to the basic layer for the scalable restoration. Again, starting from a base level $i = 1$ and with $y_{CS}^1 = \Phi_1 y = \Phi_1 D_{sc} x$ (approximately 15% of original patch image size $y_1$) we advance through enhancement layers by uniformly collecting additional number of samples (e.g., $s_2$, $s_3$, ..., $s_L = S$) in each step until the total number of $S < n$ samples is reached. Hence, given the single trained dictionary $D_{sc}$ (as in Sec. 3.1) learned over training frame for either of video test sequences, one can define an arbitrary number of sampled layers over extracted image patches.

Fig. 16 shows reconstruction results obtained via the proposed adaptive scalable CS approach averaged over the frames starting with $s_1 = 10$ and adding five more samples per each patch as frame recovery progresses (e.g., $s_2 = 15$, $s_3 = 20$, etc.). Therefore, we define in total nine sampling levels resulting in nine patch, that is, frame reconstruction layers. Thus, full number of measurements is $S = 50$ ($n > 50$) which accounts for roughly 80% of the information of the sampled signal $y_i$. The gap between the performance of the two methods is evident for the layers sampled both at low (e.g., 15%, 23%, 31% and 39%) and high subrates (47%, 55%, 62%, 70% and 80%) of sampling information, at around 3.03 [dB] in case of “Stephan” sequence and for the “Tempete” frames at 2.96 [dB]. We can see that the proposed design is successful for the subsampling factors at different rates whereas the conventional K-SVD has a comparable but not better performance as more measurements are added.

4 Discussion

Training the dictionary for the scalable sparse data representation and applying it to the denoising adopts a different approach than the one originally introduced by K-SVD [3, 9, 15]. Mainly, the atom update illustrated in Sec. 2.3 and denoising proposed Sec. 2.5 are grounded in the following assumptions:

- The progressive and quality wise scaled recovery of the image/frame can be attained via learned dictionary by modeling the main HVS perception mechanism properties and integrating them during dictionary’s training;
- This implementation should be taken forward by MCA based semi-random initialization, allocation,
separation and regularization of low and high spatial frequencies information captured by the atoms during the dictionary training procedure;

- Texture image components are less distorted by noise than the smooth ones thus with the newly introduced design SVD over proposed regularized error matrix $E_j$ is sufficient for noise removal.

These hypotheses give rise to a series of questions:

1. How are spatial frequencies distributed over scalable and non-scalable dictionary’s atoms?
2. Could this distribution be denoted as a built-in property of the trained dictionaries?
3. Does the proposed design properly adopt the HVS perception mechanism properties?
4. To what degree noise effects smooth and texture image properties?

The following sections aim to look into some answers to these questions.

4.1 Spatial frequencies distribution

In Sec. 2.1 we gave a detailed explanation on semi-random dictionary initialization where we enforce allocation and separation of the dictionaries atoms into smooth and texture ones. As explained, the classification criteria we use is formulated via Activity norm in [3]. Thus, we further assess the spatial frequencies distributions for both dictionaries by looking at and analyzing Activity trend once the training is done. This is illustrated with Fig. 17. Whether we consider frames of the video sequence (Fig. 17a and Fig. 17b) or some conventional image (Fig. 17c and Fig. 17d) we can conclude that classical K-SVD scheme results in dictionaries which do not show any specific structural features in terms of how smooth and texture information are learned and allocated. In contrast, proposed design shows clear distinction between atoms that carry:

- Low spatial frequency: $Activity(d_j)_{j=1}^{K/2} \leq A = 0.27$ (first $K/2$ atoms);
- High spatial frequency: $Activity(d_j)_{j=K/2}^{K} > A = 0.27$ (last $K/2$ atoms);

thus successfully implementing this specific distribution as a built-in property of the scalable dictionary $D_{sc}$ unlike the classical K-SVD scheme.

4.2 Contrast variation

Proper integration of the HVS sensitivity properties is done adequately if the proposed scalable design re-inforces learning of the spatial high-frequency components (see Sec. 2.3) which represent regions of a high contrast variation [33, 34]. By examining in what ways atoms of $D$ and $D_{sc}$ differ in terms of their composition
structure (i.e., contrast variation) we verify the credibility of HVS properties modeling. This is taken forward by estimating contrast levels captured within the atoms during dictionary learning procedure. Assessment of the contrast levels is done by finding the standard deviation \((\text{std})\) over the atom’s pixel intensity. Estimates are averaged over several dictionaries trained over the frames of the same video sequence or several times over the same image. The proposed computation is adopted from [37] where authors use \(\text{std}\) as a measure to estimate contrast image levels. Likewise, in Fig. 14 we depict standard deviation values for contrast levels of atoms both from \(D\) and \(D_{sc}\) for the same set of data as in the Sec. 3.1. We can notice distinct pattern for the contrast levels of the scalable “SC K-SVD” dictionary for all results likewise for Activity values shown in Sec. 4.1. Specifically, for the first \(K/2\) atoms of each of the presented scalable dictionaries \(D_{sc}\) the contrast is considerably lower with some slight fluctuations (Fig. 18, notation “SC K-SVD”) with highest contrast variation of 0.06. The remaining atoms reach quite high contrast levels with a steep jump up to around 0.13, creating a distinct threshold in distributed contrast variation over the all four \(D_{sc}\) dictionaries. The clear contrast variation borderline which clearly splits atoms in two groups, e.g., those with low and those with high contrast variation, is the final processing effect of the enforced semi-random initialization and regularization. In case of the conventionally K-SVD i.e., “NSC K-SVD” shown trend does not exist. Thus, this directly proves that proposed design complies with the characteristic of the HVS perception mechanism [29, 30] given that it is more efficient in extracting contrast information from the training images. In addition, this is significant since a proper visual understanding of the scene at hand [31, 32, 34] depends on how well contrast variations are captured with the image representational elements, that is atoms.

4.3 Noise distortion of the smooth and texture image patches

We posed an assumption in Sec. 2.5 that noise effects more smooth than texture image components. Specifically, oscillatory components of the scene i.e., texture exhibit regularity in terms of the frequency content that repeats to some extent over the image. Thus, noise which represents random signal (without any consistency in its change) should have a higher impact on image parts which do not exhibit periodic spatial variations i.e., smooth one. This is shown by estimating changes in \(\text{std}\) variation before and after noise is added to specific image blocks of smooth and texture areas. Several of these blocks are depicted in Fig. 19 where first row represents smooth and second texture image blocks of size \(30 \times 30\). Given the five noise levels as in Sec. 3.3, in Tab. 9 we show how averaged \(\text{std}\) of the texture and smooth image patches varies before and after noise is added. Given the smooth group we can see rel-
Figure 19. Visual overview of the image patches size 30x30 used for std noise impact analysis. First row represents smooth image content while seconds one depicts texture.

### Table 9. Std variation assessment averaged over group of smooth and texture image blocks size 30 × 30.

|       | Smooth | Texture |
|-------|--------|---------|
| σ = 0 | 6.67   | 46.42   |
| σ = 20| 21.42  | 50.11   |
| σ = 40| 35.82  | 56.85   |
| σ = 60| 57.59  | 67.21   |
| σ = 80| 70.44  | 76.40   |
| σ = 100| 80.22 | 84.47   |

ative jumps of 14.75, 29.15, 50.92, 63.77, 73.55 for each noise level from the initial noise free level of std 6.67. In contrast, for texture areas this change is not that steep starting from noise free std of 46.42 with relative changes of 3.69, 10.43, 20, 79, 29.89 and 38.05. Conclusion follows that noise disturbs original smooth image content on a much larger scale than the texture areas.

### 5 Conclusion

This work introduces a design for learning dictionary for scalable image recovery by enforcing semirandom dictionary initialization and regularization of the K-SVD atom’s update stage. To the best of our knowledge this problem has not been addressed before i.e., creating learned sparse representations for the purpose of the scalable image restoration. The proposed technique is evaluated over two different video test sequences, “Stephan” and “Tempete” and several conventional images. We demonstrate its practical utilization for dynamic data changing over time given single trained dictionary $D_{sc}$. Mainly, three potential applications schemes are tested: scalable video recovery, scalable denoising and scalable CS. Interestingly, the proposed approach for learning dictionary for scalable image recovery, significantly outperforms or it is highly comparable with the classical K-SVD setting for the all aforementioned purposes: (i) best for 11.32 [dB]; (ii) achieves comparable results with best decreased computational demands for 7.3 times; (iii) best for all subsampling levels at around 3.03[dB]). Future work will address joint design and optimization of the scalable CS sampling matrix and dictionary for scalable image recovery aiming for considerably decreasing current coherence level (i.e., 7.7). Overall, the proposed method can be successfully used for real-time scalable image/video display applications, where video streams, tailored to the needs of a divers user pool operating heterogeneous display equipment, are required.

### REFERENCES

[1] B. A. Olshausen and D. J. Field, “Sparse coding with an overcomplete basis set: A strategy employed by v1?,” Vision Research, 37, 1997.

[2] A. M. Bruckstein, D. L. Donoho, and M. Elad, “From sparse solutions of systems of equations to sparse modeling of signals and images,” SIAM Review, vol. 51, no. 1, pp. 34-81, 2009.

[3] M. Elad, “Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing,” SPRINGER, 2010.

[4] R. Rubinstein, A.M. Bruckstein, and M. Elad, “Dictionaries for sparse representation modeling,” Proceedings of IEEE, Special Issue on Applications of Compressive Sensing and Sparse Representation, vol. 98, no. 6, pp. 1045–1057, June 2010.

[5] D. L. Donoho, “Compressed sensing,” IEEE Trans. on Information Theory, vol. 52, pp. 1289-1306, Apr. 2006.

[6] G. Davis, S. Mallat, and Z. Zhang, “Adaptive time-frequency decompositions,” Opt. Eng, vol. 33, no. 1, pp. 2183–2191, Jan. 1994.

[7] T. Cai and L. Wang, “Orthogonal matching pursuit for sparse signal recovery with noise,” IEEE Trans. Inform. Theory, vol. 57, no. 7, pp. 4680–4688, 2011.

[8] D. L. Donoho, Y. Tsaig, and J.L. Starck, “Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit,” Technical Report, March 2006.

[9] T. T. Do, L. Gan, N. Nguyen, and T. D. Tran, “Sparsity adaptive matching pursuit algorithm for practical compressed sensing,” in Proc. of The 42th Asilomar Conference on Signals, Systems, and Computers, pp. 581-587, Oct. 2008.

[10] M. Aharon, M. Elad and A. Bruckstein, “K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation,” IEEE Trans. Im. Proc., vol. 54, pp. 4311–4322, Dec. 2006.

[11] H. Lee, A. Battle, R. Raina, and A.Y. Ng, “Efficient Sparse Coding Algorithms,” Advances in Neural Information Processing Systems, MIT Press, pp. 801-808, 2007.
[44] J. M. Duarte-Carvajalino and G. Sapiro, “Learning to sense sparse signals: Simultaneous sensing matrix and sparsifying dictionary optimization,” *IEEE Trans. Image Proc.*, vol. 18, no. 7, pp. 1395-1408, July 2009.

[45] M. Wakin, J. Laska, M. Duarte, D. Baron, S. Sarvotham, D. Takhar, K. Kelly, and R.G. Baraniuk, “Compressive imaging for video representation and coding,” *PCS-2006 Picture Coding Symposium*, vol. 52, pp. 1289-1306, Beijing, China, Apr. 2006.

[46] V. Stankovic, L. Stankovic, and S. Cheng, “Compressive video sampling,” *Proc. EUSIPCO, Lausanne, Switzerland*, Aug. 2008.