Sparse molecular image representation

Sofia Karygianni *, Pascal Frossard

Ecole Polytechnique Fédérale de Lausanne (EPFL), Signal Processing Laboratory (LTS4), CH-1015 Lausanne, Switzerland

A R T I C L E   I N F O

Article history:
Received 8 June 2015
Accepted 25 January 2016
Available online 2 February 2016

Keywords:
Sparsity
Structure
Linear combinations
Molecules
Prototypes
Deformations
Pools
Structural similarity

A B S T R A C T

Sparsity-based models have proven to be very effective in most image processing applications. The notion of sparsity has recently been extended to structured sparsity models where not only the number of components but also their support is important. This paper goes one step further and proposes a new model where signals are composed of a small number of molecules, which are each linear combinations of a few elementary functions in a dictionary. Our model takes into account the energy on the signal components in addition to their support. We study our prior in detail and propose a novel algorithm for sparse coding that permits the appearance of signal dependent versions of the molecules. Our experiments prove the benefits of the new image model in various restoration tasks and confirm the effectiveness of priors that extend sparsity in flexible ways especially in case of inverse problems with low quality data.

1. Introduction

Most tasks in signal processing and analysis are significantly simplified when the data is represented into its right form, especially for high-dimensional signals like images. The quest for the right signal representation has fostered the use of overcomplete dictionaries as tools for signal compression, denoising, enhancement and various other applications. Dictionaries have the advantage to have very few constraints in their construction, so that they can be finally adapted to the data processing task at hand. However, this flexibility has a price: the representation of a signal is unfortunately not unique in overcomplete dictionaries, and finding the best such representation is generally an ill-posed problem. As a result, well-chosen priors or models about the signal representation become necessary in order to develop effective signal processing algorithms with overcomplete representations.

The most common models in overcomplete signal representations are based on sparsity priors. This means that the signal is well represented by only a few components or atoms of the overcomplete dictionary. Sparsity is a pretty intuitive prior that is also biologically plausible, as shown in the pioneer work of Olshausen and Field [1] where it is suggested that sparsity could be a property employed by the mammalian visual system for achieving efficient representations of natural images. Vast research efforts have been deployed in the last decades in order to design algorithms that solve the hard problem of sparse decomposition of signals by effective approximation [2,3] or convex relaxation [4,5].

While sparsity is a simple and generic model, it is not always a sufficient prior to obtain good signal reconstruction, especially if the original data measurements are compressed or inaccurate. More effective signal models can therefore be built by considering the dependencies between the dictionary elements that appear in the signal representation instead of their number only. In that spirit, group sparsity has been introduced as a way to enforce a pre-specified structure in the decomposition. Specifically, the components of the dictionary are partitioned into groups and the elements of each group are encouraged to appear simultaneously in the signal decomposition [6]. Alternatively, the atoms can also obey a predefined hierarchical structure [7]. Other approaches have considered additional flexibility by constraining the signal decomposition to include elements from overlapping groups of atoms [8–10]. The group sparsity structure is however not always appropriate for modeling signal patterns as the groups are merely identified in terms of their support. It is however not suitable for differentiating patterns with the same support but different distributions, which could actually be very different signal patterns. Such a case is presented in Fig. 1 where we show how much the image of a face can change when varying the coefficients of its sparse code while keeping the same support. This ambiguity is unfortunately a serious drawback in various applications such as signal recovery and recognition, for example.

We propose here a new signal model to represent signal patterns and higher level structures. Our goal is to build richer priors
than classical structured sparsity models that merely focus on the support of the signal representation and not on the actual energy distribution. Our model builds on our previous work on structured sparsity [11] and represents signals as sparse sets of molecules, which are linear combinations of atoms from a redundant dictionary of elementary functions. To enhance the flexibility of our model, in this work we go one step further and instead of allowing only small variations in the coefficients of the molecules, we allow molecule realizations to appear in various forms that can have small deviations on both their coefficients and their support. To this end, we form pools of similar atoms in the dictionary, and assume that all atoms in a pool carry similar information. The molecule realizations are then defined as slightly deformed versions of the molecule prototypes, where atoms could be replaced by similar atoms from their respective pools. As a result, a given molecule prototype represents a group of structurally similar patterns whose exact form in signals is controlled by the construction of the atom pools. This provides flexibility in the representation of signals with molecules, while preserving the main structural information in the sparse signal approximation.

We study in details our new structured sparsity model and analyze the recovery performance of molecule representations. We formally show that our choice of the synthesis dictionary based on molecules realizations provides a good compromise between structure and flexibility. Then we propose a novel constructive sparse coding algorithm of signals with our new structured sparsity model. We exploit the characteristics of atoms pools to design effective similarity measures for detecting molecule realizations in signals. Finally, we show the use of our new framework with illustrative experiments in various applications such as compressed sensing, inpainting and denoising. Our results show that the new structured sparsity prior leads to better reconstruction performance than classical sparsity priors due to its flexible molecule-based representation.

Our efficient structured sparsity model represents a quite unique framework in the literature. In particular, the consideration of the coefficient distribution and the atom pools, as well as the definition of both molecule prototypes and realizations, are important characteristics of our new signal representation model. The coefficients permit to differentiate structures with distinct energy distributions on the same support and thus to facilitate the proper recovery of image information in case of incomplete or inaccurate observations. Another definition of molecule has been previously proposed in [12] to describe a set of coherent atoms in a dictionary, but it is more related to the notion of a group or a pool of atoms than to our original definition of a molecule. Multi-level structures are also related to the concept of double sparsity introduced in [13] where the authors learn structures on top of a set of predefined set of atoms. It is however less flexible than our model, where we include the notion of pools and molecules realizations that enable the proper handling of minor structure deformation in the signals. Less close to our model, some recent works describe the statistical dependencies between the atoms in a dictionary with graphical models. For example, Markov Random Fields (MRFs) are employed for modeling these dependencies in [14–16]. The resulting structure model is a probability distribution function that compares the different possible supports of atoms in the signal representation. These models are quite powerful but unfortunately quite complicated and highly parametric, such that they are difficult to deploy and adapt to various applications. Next, the idea of pooling that is used for defining molecules realizations is quite often used under different forms to provide local invariance [17,18] in the signal representation. In our case however, it provides local invariance to small deformations of a set of atoms with higher resilience to sparse code variability in the identification of typical patterns in images. Finally, the differentiation between the molecule prototypes and molecule realizations in our new model leads to realizations of structures that are signal dependent, like in [19,20]. Hence, the signal representation is flexible but nevertheless follows a pre-defined structure. The specific characteristics of our scheme make it very suitable for various signal processing tasks and especially signal denoising and inpainting.

The structured sparsity model proposed in this paper is essentially a two-layer architecture with the first layer consisting of the dictionary atoms and the second of the molecules. The benefits of such architectures over the flat ones has been a subject of research for a long time in the feature extraction and machine learning community. It has been validated experimentally in the case of signal recognition in [21] while the mere existence of the field of deep learning can argue in benefit of multistage architectures. The deep learning systems consist of a hierarchy of features along with some pooling and contrast normalization operators that
sequentially transform the input into a new representation [18,22,23,20]. Although it is common to learn the filters used in each layer from the data, there is recent work done also in the case of predefined filters [24]. In both cases, the goal of the learning is to uncover class invariant signal representations that are mainly used for classification and not the learning of appropriate structure signal priors for signal recovery. These works nevertheless support the idea that multiple layers leads to better signal models, which is aligned with the ideas proposed in this paper.

To summarize, the main contributions of our work are:

- We propose and study a novel two-layer signal model built on atoms and respectively molecules, where the inclusion of both the coefficients and the support in the structured sparsity prior permits the differentiation of structures on the same support but with distinct energy distributions.
- We define the atom pools to permit the differentiation between molecule prototypes and molecule realizations in order to be resilient to variability in the sparse codes.
- We design a new algorithm for sparse coding under our new structured sparsity prior, which achieves very promising results in illustrative signal restoration tasks.

The rest of the paper is organized as follows. In Section 2 we describe our model in detail. In Section 3 we compare the different choices for a suitable synthesis dictionary for our model while in Section 4 we present the associated coding problem in detail. Finally, in Section 5 we provide results that validate our model for various signal restoration tasks.

2. Structured image model

2.1. Multi-level structure

We present now our new structured sparsity model for images whose multi-level structure permits to represent visual patterns or typical signal parts as combinations of elementary atoms in a dictionary. In other words, we define molecules as linear combinations of atoms to represent groups of structurally similar signal patterns. We further differentiate the molecules into molecule prototypes and molecule realizations, which are slightly deformed versions of the prototypes aiming at capturing signal variability. We first present our new model and then discuss in details the notion of pools of atoms, which is central for computing molecule realizations. Then we introduce a new structural difference function that is later used to compare visual patterns based on the value of their coefficient vectors.

To start with, we first provide an example to illustrate our structured sparsity model. Our model is built on the concepts of molecule prototypes and realizations. The prototype is a representative pattern for a group of molecule realizations, which are slightly deformed versions of a typical image part. The first image in Fig. 2a shows a molecule prototype, which is an orthogonal angle formed by two edge-like atoms from the dictionary of elementary atoms. In other words, the molecule prototype is represented by a particular linear combinations of atoms, as shown in the first energy distribution function in Fig. 2b. The molecule could however appear with small deformations in actual images, and such molecules realizations are illustrated in the rest of the images in Fig. 2a. They look quite similar to the molecule prototype and preserve to some extent its structural characteristics, but they are not constructed with the exact same atoms, as illustrated by their respective energy distribution functions in Fig. 2b.

We know describe our new signal model in more details. We consider a set of signals \( X \in \mathbb{R}^{N \times D} \) and a base dictionary \( D \in \mathbb{R}^{N \times K} \) of elementary functions or atoms \( d_k \) with \( 1 \leq k \leq K \), whose linear combinations can effectively represent the signals \( X \). We assume that the occurrence of atoms in the signal representation is not completely independent but that atoms rather have the tendency to form typical visual patterns. In other words, there are some linear combinations of atoms that tend to appear more frequently than others, possibly with slight changes either in the energy distribution or atom sets. The most frequent atom combinations are represented by a set of molecule prototypes \( M = \{ m_l, l \in \{ 1, \ldots, Q \} \} \) where each prototype is defined as a sparse set of atoms with specific coefficient values, i.e.,

\[
m_l = \sum_{k=1}^{K} c_{z,l}(k)d_k = DC_{z,l}, \quad \|c_{z,l}\|_0 < n \tag{1}
\]

where \( n \) is the sparsity level of the molecules and \( c_{z,l}(k) > 0 \) only if the atom \( d_k \) belongs to the support \( \mathcal{I}_{z,l} \) of the molecule \( m_l \). The non-negativity of coefficients will be explained in more detail in Section 2.2. We can further write all the molecule prototypes in a matrix form as

\[
M = DC_z, \quad \text{with} \; C_z = [c_{z,1} c_{z,2} \cdots c_{z,Q}]. \tag{2}
\]

We consider that the molecules correspond to the most important parts in the signals, but that they may appear as realizations that are similar but not identical to the prototypes. Equivalently, we consider a signal \( x \in X \) to be a sparse non-negative combination of molecule realizations plus some bounded noise. We define \( c_{x,l} \) as the vector of atom coefficients that expresses the realization of the molecule \( m_l \) in \( x \). We further consider that the difference between a molecule realization and the corresponding prototype is small, i.e., \( A(c_{x,l}, c_{z,l}) < t, \forall l \) where the function \( A \) measures the structural difference between molecules. The parameter \( t \) is a threshold value on the structural difference and its value permits to control the flexibility of our new multi-level model in capturing the variability in typical visual patterns. The signal can therefore be written as

\[
x = DC_{x,a} + \eta, \quad \text{with} \; C_x = [c_{x,1} c_{x,2} \cdots c_{x,Q}] \quad \text{and} \quad A(c_{x,l}, c_{z,l}) < t, \forall l \tag{3}
\]

We further consider that the approximation error is bounded (i.e., \( \|\eta\|_2 < H \)), the atom and molecule coefficients are defined as \( a_l \geq 0, \forall l \) and \( c_{x,l}(k) \geq 0, \forall (k,l) \) and the representation is sparse, i.e., \( \|a_l\|_0 \leq s \) for some sparsity threshold \( s \).

The image model in Eq. (3) corresponds to a sparse decomposition of \( x \) into molecule realizations, or equivalently the expansion of the signal \( x \) into dictionary atoms whose coefficients are given by \( C_{x,a} \). The grouping of atoms into molecules is driven by the choice of the structural difference function \( A \) that quantifies the deviation of molecule realizations from the corresponding prototypes. A graphical illustration of the newly introduced concept of molecules is provided in Fig. 3. In the rest of this section, we first introduce the concept of atom pools, which are groups of similar atoms in the dictionary, and eventually use atom pools to define the structural difference metric \( A \) that is used in our molecule-sparse signal model.

2.2. Pools of atoms

In our framework, the signal is represented as a linear combination of atoms taken from a redundant dictionary. The redundancy of the dictionary helps in building sparse representations but also leads to the fact that many atoms may carry similar information. In
particular, a specific image feature can be well captured by a specific atom in the dictionary. But the same feature might also be well represented by atoms that are similar to , as illustrated in Fig. 4. Depending on the actual image representation method, the same visual feature can therefore be coded in various ways. We would like to make sure that our part-sparse image model is able to take this phenomenon into account.

We define the notion of atom pools in order to represent atoms that are similar and that have similar contributions in a molecule. More specifically, in a dictionary , each atom can be represented as a unit norm vector in the signal space . To measure the similarity between the atoms we use the cosine similarity, defined as

\[
\langle d_i, d_j \rangle_{\mathbb{R}^N} = \frac{d_i^T d_j}{||d_i|| ||d_j||}
\]

where \(\langle d_i, d_j \rangle\) is the dot product between the atoms \(d_i\) and \(d_j\) and \(||d_i||\), \(||d_j||\) correspond to their \(l_2\) norm. Under the assumption that the atoms are normalized, the cosine similarity between \(d_i\) and \(d_j\) equals their dot product. If the dot product between two atoms \(d_i, d_j\) is very high, the energy of the projection of \(d_j\) on the direction of \(d_i\) is significant, so that a visual feature may be equivalently well represented by the atoms \(d_i\) or \(d_j\). We characterize this phenomenon by introducing the notion of pools of atoms: each atom \(d_i\) is related to a pool \(P(d_i)\) of atoms \(d_j\)'s that are similar to \(d_i\). In other words, a pool is defined as

\[
P(d_i) = \{d_j, 1 \leq j \leq K, \ | \langle d_i, d_j \rangle | > 1 - \epsilon \}
\]

where \(\epsilon\) is a suitable chosen threshold depending on the application at hand and the coherence of the underlying dictionary \(D\).

Fig. 2. Illustrative example of a molecule prototype and its realizations. In (a) the molecule prototype (on the left) represents a near orthogonal crossing of edges while the molecule realizations describe visual patterns that are similar to the prototype. The \(l_2\) distance between the prototype and the realizations in the image domain is given on top of each realization. In (b) we show the corresponding sparse codes of the images in (a). The \(l_2\) distance of the sparse codes seen as vectors in \(\mathbb{R}^N\) is given on top of each figure. As we can see, none of the metrics depicts accurately the structural similarity among the patterns.

Fig. 3. Multilevel representation of atoms and molecules. In the first level we have the atoms whose similarities, captured by the atom pools, are represented as links between the corresponding atoms. In the second level, we have the molecules which are linear combinations of atoms. The molecules are further distinguished into molecule prototypes which are the main signal patterns to be represented and molecule realizations that are molecules that are structurally similar to the prototypes.
Equipped with this definition, we can now measure the difference between alternative representations of the same visual features. In particular, we can estimate the actual energy corresponding to the atom \( d_i \) in a signal represented by the sparse code \( b \) that does actually not include the atom \( d_i \). In other words, looking at the sparse signal decomposition \( x = Db \) with \( b_i = 0 \), we would like to know how much of the energy is actually aligned along the direction represented by the atom \( d_i \). It mainly corresponds to the energy captured by the coefficient of all the atoms in the pool \( P(d_i) \). We can therefore approximate the energy of the signal in the direction of \( d_i \) as

\[
\epsilon_i(b) = \sum_{j \in P(d_i)} b_j(d, d_j) = S_i b
\]

where

\[
S_i(j) = \begin{cases} 
\langle d_i, d_j \rangle & \text{if } d_j \in P(d_i) \\
0 & \text{if } d_j \notin P(d_i) 
\end{cases}
\]

The vector \( S_i \) expresses essentially the pairwise relationships between the atom \( d_i \) and the rest of the atoms in the dictionary \( D \). The energy estimate above is very useful in computing the structural difference between molecules that is explained below. The value of \( \epsilon_i(b) \) is essentially the length of the projection of the vector \( \nu_i(b) = \sum_{j \in P(d_i)} b_j d_j \), the realization of \( d_i \), in the direction of \( d_i \). When the entries of \( b \) are non-negative, \( \nu_i \) is guaranteed to lie in the geometric space defined by the pool \( P(d_i) \) and as a result the error \( \|d_i - \nu_i\|_2^2 \) is bounded (the proof is provided in Appendix A). In the rest, we will adopt this assumption of non-negativity without loss of generality. Finally, an example of the pool of an atom, as well as a possible non-negative realization of the atom from its pool, is shown in Fig. 4.

### 2.3. Structural difference

We now propose a measure of structural difference between molecule instances that is based on the above definition of atom pools. Recall that a molecule realization is similar to a molecule prototype and permits to capture the variability of visual patterns in actual images. It can be defined as the deformation of a molecule prototype whose original atoms could be each substituted by atoms from their respective pool. Equivalently, a molecule realization is essentially a molecule prototype that can be realized through a linear combinations of atoms in the pools of the initial prototype components. As a result, a molecule realization has a similar energy as the prototype when measured on atom pools but not necessarily exactly the same coefficient values on the atoms. It makes it difficult to measure the similarity between the actual visual patterns represented by the molecule prototype and its realizations. For example, the \( l_1 \) norm in both the image and sparse code domain fail to uncover the structural similarity between both molecules, as it does not take into account the actual features represented by the atoms nor their interplay. The inability of the \( l_1 \) norm in capturing the similarity of molecules can be observed by checking the norms in Fig. 2a and b.

As classical norm metrics are not appropriate for computing the similarity in the structure of different molecule instances, we propose a new structural difference \( A \) for the signal model of Eq. (3). In particular, the deformation in the structure of molecules is measured by the compatibility between a sparse coefficient vector \( c_{x,i} \) that represents the realization of the molecule \( m_i \) in the signal \( x \), and the sparse coefficient vector \( c_{x,j} \) that represents the corresponding molecule prototype. Since a molecule is identified by specific energy levels on the pools of the atoms in its support, its realizations are allowed to have non-zero values only in the union of the pools of these atoms, i.e., \( \Gamma_{x,i} \subseteq \bigcup_{k \in \Gamma_{x,i}} P(d_k) \) where \( \Gamma_{x,i} \) and \( \Gamma_{x,j} \) are the supports of \( c_{x,i} \) and \( c_{x,j} \), respectively. Then, the structural difference computes the energy in the pools of \( c_{x,j} \) and compares it to the ones expressed in \( c_{x,i} \). If the energies are comparable, the structural difference is considered to be small.

To be more specific, using the formula for the energy level of an atom based on its pool given in (5), the structural difference \( A \) is computed as:

\[
A(c_{x,i}, c_{x,j}) = \sum_{k \in \Gamma_{x,i} \cap \Gamma_{x,j}} (c_{x,i}(k) - c_{x,j}(k))^2
\]

\[
= \sum_{k \in \Gamma_{x,i} \cap \Gamma_{x,j}} (c_{x,i}(k) - S c_{x,j})^2 = \|W_l \times (c_{x,i} - S c_{x,j})\|_2^2
\]

where \( S = [S_1, S_2, \ldots, S_k] \), with \( S_i \) from Eq. (6). The indicator vector \( W_l \) denotes the inclusion of dictionary atoms in the support \( \Gamma_{x,i} \) of the molecule \( m_i \), i.e.,

\[
W_l(k) = \begin{cases} 
1 & \text{if } d_k \in \Gamma_{x,i} \\
0 & \text{if } d_k \notin \Gamma_{x,i}
\end{cases}
\]

Note that atoms that participate in the same molecule are assumed to not have overlapping pools which is equivalent to assuming that the atoms in a prototype are quite incoherent. As we will see in Section 3 this is a desired property that leads to lower coherence on the dictionary and thus better recovery guarantees. In general, the lower the structural difference \( A(c_{x,i}, c_{x,j}) \), the more compatible the molecule realization and its prototype. Finally, we show an example of a molecule prototype and one possible realization in the atomic level in Fig. 5 along with the corresponding structural difference function.

### 3. Recovery analysis

The proposed model presented in Eq. (1) defines signals to be formed as a composition of molecule prototypes with small, controlled deformations. The molecules are further defined as linear combinations of a set of basic atoms. According to this model, one could approximate signals in three different ways, namely as linear combinations of elements in three different dictionaries: the atomic dictionary \( D \), the molecule prototype dictionary \( DC \) and the dictionary of molecule realizations. In the rest of this section, we analyze the pros and cons of each option in accurately representing signals.
Therefore, it appears that building a dictionary with all possible molecule realizations, denoted as $DC_r$, could be a better and more flexible alternative with a compromise between structure and flexibility. However, building a dictionary with all possible molecule realizations results in a very coherent representation. As we have seen in Section 2.1, the molecule realizations are essentially small deformations of a molecule prototype. Therefore, all realizations of the same prototype are highly similar. The recovery performance of a dictionary is known to deteriorate as the sparsity of the signals decreases and the coherence of the dictionary increases. To put it more formally, a known recovery constraint for BPDN (Basis Pursuit Denoising) [25] or OMP (Orthogonal Matching Pursuit) [3] is given by

$$k \leq \frac{1}{2} \left( \frac{1}{\mu_r} + 1 \right).$$

(9)

where $\mu_r$ is the coherence of the underlying dictionary and $k$ is the sparsity of the signal, i.e., the number of elements in the signal. Therefore, the more coherent the dictionary $DC_r$, the more sparse the signals should be in order to be able to recover them.

We can analyze how the coherence $\mu_r$ of the dictionary $DC_r$ is affected by the presence of multiple realizations for each molecule prototype. Since the realizations of the same molecule prototype are very similar, $\mu_r$ can be lower bounded using the maximum distance $r$ between any realization and the corresponding molecule prototype. The theoretical bound, $L_x$, is given by

$$L_x = 1 - 2r^2$$

(10)

To quantify this result, we can compare the molecule realization dictionary with the case of a dictionary $DC_u$ that contains only one molecule realization per molecule prototype. The restriction on the allowed number of instances per prototype allows for a theoretical upper bound on the coherence $\mu_u$ of the dictionary $DC_u$, i.e., $U_u \geq \mu_u$ with

$$U_u = \mu(1 - 2r^2) + 2r\sqrt{(1 - \mu^2)(1 - r^2)}$$

(11)

where $\mu$ is the coherence of the dictionary of molecule prototypes $DC$. In practice the coherence $\mu_u$ is expected to be close to $\mu$. Both theoretical bounds depend on the distance $r$ which is driven by the characteristics of the atoms pools as well as the internal structure of the molecules. The latter is measured by the maximum similarity between atoms belonging to the same molecule, denoted as $\mu_{sr}$. To improve the readability of the section we have moved the exact expressions for $r$ as well as the proofs for the bounds in Appendix B.

From the expression for $L_x$ we can see that the smaller the $r$ is, the worse the $\mu_u$ is expected to be. On the other hand, when $r$ is small, $U_u$ gets closer to $\mu$. In order to present these dependencies more concretely, we show in Fig. 6 some plots of $\mu_u$ and $\mu_r$ for various settings. At the first row, we present the bounds $L_x$ and $U_u$ computed based on Eqs. (10) and (11) respectively while at the second row we show the mean values of $\mu_u$ and $\mu_r$ computed experimentally for different values of the molecule prototype coherence $\mu$ over random generations of the dictionaries $DC_u$ and $DC_r$. For simplicity, in our calculations we have assumed that the number of atoms in all molecules is the same, denoted as $n$. The pool angle $\phi$ was set to 10 degrees while we varied the maximum in-molecule atomic similarity $\mu_{sr}$. In both rows, the red line refers to the coherence of the $DC_r$ dictionary, the blue line to the coherence of $DC_u$ and the yellow to that of molecule prototypes $DC$.

---

**Fig. 5.** Illustration of a molecule prototype and a possible realization. The vector $W_r$ is the indicator function of the support $G_x^{\ell}$ of the molecule prototype $c_x$. The structural difference between $c_x$ and $r_x$ is then

$$d(c_x, r_x) = ||W_r \times (c_x - S_x)||_2^2 = (c_2 - (d_1, d_2) b_1)^2 + (c_21 - b_21)^2 + (c_46 - (d_46, d_47) b_47)^2.$$  

On the one hand, the benefit of the atomic dictionary, is its flexibility since it includes all possible atoms present in signals. However, the lack of any structure makes it less appropriate for recovering signals under challenging conditions, in the presence of intense noise or when information is missing, as the sparsity prior may prove to be insufficient for a satisfactory reconstruction. On the other hand, it is known that the inclusion of more structure in the dictionaries facilitates significantly the task of signal restoration even under severe degradation. The dictionary of molecule prototypes as well as that of molecule realizations have both the advantage of providing structured priors. However, this advantage comes at a price in both cases.

The dictionary of molecule prototypes, might not be always sufficient for recovering the right structure in the signals. We can rewrite a signal given from Eq. (1) as:

$$x = DC_x a + \eta = D(C + E_x) a + \eta \approx DC_x a + DC \tilde{a} + \eta$$

$$= DC(a + \tilde{a}) + \eta$$

$$= DC b + \eta$$

where $DC \tilde{a}$ is the best approximation of $DE_x a$ in the dictionary of molecule prototypes $DC$. The vector $\tilde{a}$ is an exact sparse representation. However, $E_x$, which is the structured deviation from the prototypes, can take various forms so that the vector $\tilde{a}$ does not necessarily have a sparse nature. Therefore, the structure of $b$ can be significantly different from that of $a$ resulting in a false recovery of the signal structure. The source of the above problem is the lack of flexibility in the dictionary $DC$ in contrast to the more versatile $DC_x$ that is a dictionary of molecule realizations and can thus take different forms by including different molecule realizations.
tional molecule prototype dictionary presented in the next section. The details of the exact coherence of the used dictionary while through the iterations we expect to recover the right signal structure. The coherence of the molecule prototype dictionary quite well while 1. Therefore, we observe that restricting the number of realizations in the dictionary to one per molecule prototype preserves the coherence of the molecule prototype dictionary quite well while the inclusion of more than one molecule realizations per prototype pushes the dictionary coherence towards 1, thus making the dictionary less suitable for sparse recovery.

To sum up, from the above discussion we can see that deciding which dictionary to use for signal decomposition is not trivial. The underlying atomic dictionary \( D \) lacks structure, the dictionary of molecule prototypes \( D_\text{C} \) lacks flexibility while the dictionary of all molecule realizations suffers from inefficient size and high coherence. To alleviate this issue, we propose an iterative decomposition scheme that searches for the best molecule realizations using at each iteration a synthesis dictionary with strictly one molecule realization per molecule prototype, denoted as \( D_{\text{C},a} \) above. In this way, at each iteration we have a guarantee for the coherence of the used dictionary while through the iterations we expect to recover the right signal structure. The details of the exact problem formulation as well as the proposed solution are presented in the next section.

4. Adaptive molecule coding algorithm

We now formulate the problem of decomposing a signal into a sparse set of molecule realizations. We assume that the signal \( x \) follows the model in Eq. (3), or equivalently that the signal can be well approximated by a sparse linear combination of molecule realizations represented by \( C_x \), along with their respective coefficients \( a \). Each molecule realization in \( C_x \) is a small deformation of the corresponding molecule prototype in \( C \). The signal approximation can then be computed by solving the adaptive molecule coding problem written as follows:

\[
\begin{align*}
\{\hat{a}, \hat{C}_x\} &= \arg\min_{a, C_x} \left[ \|x - DC_x a\|^2_2 + \lambda_1 \|a\|_1 + \sum_{l,a(i),l'} \left\{ \lambda_2 A(c_{xl}, c_{xl'}) + \lambda_3 \|C_{xl} - C_{xl'}\|_1 \right\} \right] \\
&= \arg\min_{a, C_x} \left[ \|x - DC_x a\|^2_2 + \lambda_1 \|a\|_1 + \sum_{i} \lambda_2 A(c_{xi}, C_x) + \lambda_3 \sum_{l,a(i),l'} \|C_{xl} - C_{xl'}\|_1 \right]
\end{align*}
\]

where each \( c_{xi} \) is a molecule realization for the molecule prototype \( c_{xl} \) and \( a[l] \geq 0 \), \( C_x[k,l] \geq 0 \forall l,k \). The first term in the objective function in Eq. (12) is the error of the approximation of the signal with a sparse set of molecule realizations. The second term favors a sparse approximation with the \( l_1 \) norm of the coefficient vector \( a \). The last term drives the form of the molecule realizations: the term \( A(c_{xi}, C_x) \) tends to favor molecules realizations that are close to prototypes while the \( l_1 \) norm on the molecules realizations codes \( c_{xi} \) ensures their sparsity. The weight parameters \( \lambda_i \)'s permit to balance the different terms of the objective function.

By substituting the structural difference function from Eq. (7) in Eq. (12) we get:

\[
\begin{align*}
\{\hat{a}, \hat{C}_x\} &= \arg\min_{a, C_x} \left[ \|x - DC_x a\|^2_2 + \lambda_1 \|a\|_1 + \lambda_2 \sum_{l,a(i),l'} (\|W_l \times (c_{xl} - SC_{xl'})\|_2^2) \right. \\
&\quad \left. + \lambda_3 \sum_{l,a(i),l'} \|C_{xl} - C_{xl'}\|_1 \right]
\end{align*}
\]

where \( W_l \) is given in Eq. (8) and the set of \( \lambda_i \)'s are weight parameters. For a given dictionary \( D \), a set of pools represented by \( S \) and a set of molecule prototype written as \( C_x \), the objective function \( F_{\text{DSS}} \) is minimized when the variables \( a \) and \( C_x \) form a part-sparse approximation of \( x \). However, the above optimization problem cannot unfortunately be solved easily for both variables \( a \) and \( C_x \) at the same time. Clearly, the problem is not jointly convex for both variables. However, when one of the variables is fixed, the problem is convex with respect to the other one. Therefore, we adopt an alternating optimization technique with two steps for solving the optimization problem in Eq. (13). The two steps are computed as follows.
1. We first fix the set of molecules realizations, and solve the sparse coding problem for the coefficient vector $a$. Given $C_a$, the solution for $a$ can be found as:

$$\hat{a} = \arg\min_a \left[ \|x - DC_a a\|^2_2 + \lambda_1 \|a\|_1 \right],$$

with $a(i) \geq 0, \forall i$ \hfill (14)

2. Then, we fix the coefficient vector, and find the set of molecule realizations that minimize the objective function of the coding problem. Given $a$, the solution for $C_a$ can be found as

$$C_a = \arg\min_{C_a} \left[ \|x - DC_a a\|^2_2 + \lambda_2 \sum_{i \in \phi} \|\{W_i (C_{a,i} - SC_a)\}_i\|^2_2 + \lambda_3 \sum_{i \in \phi} \|C_{a,i}\|_1 \right],$$

with $C_{a,i}(k) \geq 0, \forall i, k$. \hfill (15)

The first problem is essentially an $l_1$ regularized sparse coding problem which is convex with $a$. It can be solved with many different algorithms, e.g., [2,26]. To be able to handle cases of very high-dimensional data or very big dictionaries, we have chosen to solve it with the method of alternating direction method of multipliers (ADMM) [27] which is a method suitable for large scale problems. Following the findings in [28], we also employ the method of reweighted $l_1$-minimization as it leads to a sparser solution. Note that, at the very first iteration of the global algorithm, $C_a$ is initialized with $C$, while it is later updated during the solution of the second step of the alternating algorithm.

The second problem is also convex. As for the first problem, we have chosen to solve it with ADMM [27]. In order to solve it more efficiently, we however rewrite it so that it is optimized for one vector of coefficients $b$ instead of the matrix $C_a$. As explained in Section 2.3, the support of each molecule realization is restricted to the union of the pools of the active atoms in the corresponding molecule prototype. Therefore, many of the entries in matrix $C_a$ are constrained to be zero. The vector $b$ represents the possible non-zero entries in $C_a$, i.e. the coefficients of the atoms that are part of the pools of the active atoms in the molecules that compose $x$ (given in $a$). Essentially it expresses the flexibility that is allowed in the molecule realizations once the molecules are chosen.

To complete our problem transformation, we further introduce the vector $\tilde{C}$ that expresses the expected energy in the atoms pools. It is created by concatenating into a vector form the entries in $C$ that correspond to the energy expected in each pool of active atoms. Then, we also create a corresponding dictionary of atoms $\tilde{D}$ by concatenating the atoms in each of the active pools. Finally, the new vector of relationships $\tilde{S}$ between atoms in $\tilde{D}$ replaces the vector $S$. With these modifications, the problem in Eq. (15) can be equivalently expressed as:

$$\tilde{b} = \arg\min_{\tilde{b}} \left[ \|x - \tilde{D} \tilde{b}\|^2_2 + \lambda_2 \|\tilde{C} - \tilde{S} \tilde{b}\|^2_2 + \lambda_3 \|\tilde{b}\|_1 \right],$$

with $b(k) \geq 0, \forall k$. \hfill (16)

The solution of this problem is much less time consuming than the one of the equivalent problem in Eq. (15) as the size of the vector $b$ is usually much smaller than that of the whole dictionary $D$.

Finally, we iterate between the two optimization problems until the value of the signal reconstruction does not change much. Although this alternate optimization technique does not have any optimality guarantee, it gives good results in practice and therefore offers an effective constructive solution to the sparse coding problem of Eq. (13). Since the algorithm has several constraints on the structure and sparsity the final molecule realizations cannot be completely different from the predefined molecule prototypes and as a result the quality of the signal reconstruction depends significantly on the initialization of the molecule structure. However, the design and learning of good molecule prototypes is beyond the scope of this paper which is mainly focused on the sparse coding step and remains as interesting future work.

Moreover, even though the proposed adaptive coding scheme is computationally more demanding than simple sparse coding, its complexity is not very high and certainly reasonable. This is true because each sub-problem is solved with ADMM [27], which is suitable for large scale problems; thus it is quite efficient even with very high-dimensional data or very big dictionaries. Moreover, in our case, the internal computations performed by ADMM are very efficient as the intermediate update steps of the involved variables have a closed form expression. What’s more, with the transformation that we propose for the second sub-problem of the alternate optimization in Eq. (16), the computational complexity is reduced significantly due to the decrease on the size of the dictionary.

Finally, as far as the parameters of the algorithm are concerned, the values for the $\lambda$’s for each specific task are chosen based on a small validation set. As a general rule of thumb, the parameter $\lambda_2$ should be set equal or higher than the others, as it is the one controlling the importance of the structural difference between the molecule prototypes and their realizations. In case $\lambda_2$ is set to 0, the problem becomes equivalent to ordinary sparse coding. The values of $\lambda_1$ and $\lambda_3$ depend a lot on the size of the corresponding molecular and atomic dictionaries, with the most overcomplete dictionary usually requiring a higher parameter value. Finally, in our experiments the value for the parameter $r$ required for the ADMM method is set to 1, which is the value that provided the best results on our validation tests. The pseudocode of the complete sparse coding scheme, called Adaptive Molecule Coding (AMC), is presented in Algorithm 1.

**Algorithm 1. Adaptive molecule coding (AMC)**

**Input:** $x, D, C, \sigma, x, r, \lambda_2, \lambda_3, \epsilon$
1: \(\hat{a} = \arg\min_a \|x - DC_a a\|_2 + \lambda_2 \|a\|_1 \), $a \geq 0$ \hfill (17) \hfill \begin{itemize} \item \texttt{Initialize } $a$ \end{itemize}
2: while true do \hfill \begin{itemize} \item \texttt{Alternate optimization} \end{itemize}
3: \(D, S, C = \text{transform}(D, C, S, \hat{a}) \) \hfill (16) \hfill \begin{itemize} \item \texttt{Create new variables for } \text{Eq. (16)} \end{itemize}
4: \(\tilde{b} = \arg\min_{\tilde{b}} \left[ \|x - \tilde{D} \tilde{b}\|^2_2 + \lambda_2 \|\tilde{C} - \tilde{S} \tilde{b}\|^2_2 + \lambda_3 \|\tilde{b}\|_1 \right], \ b \geq 0 \) \hfill \begin{itemize} \item \texttt{Solve for } $\tilde{b}$ \end{itemize}
5: \(\hat{C}_a = \text{transform}^{-1}(\tilde{b}, C, \hat{a}) \) \hfill \begin{itemize} \item \texttt{Reconstruct } $C_a$ \text{ from } $\tilde{b}$ \end{itemize}
6: \(w = 1 / \hat{a} \) \hfill \begin{itemize} \item \texttt{Set new weights for re-weighted } $l_1$ \end{itemize}
7: \(\hat{a} = \arg\min_a \left[ \|x - \hat{C}_a a\|_2 + \lambda_1 \|w \ast a\|_1 \right], \ a \geq 0 \) \hfill \begin{itemize} \item \texttt{Solve for } $a$ \end{itemize}
8: if $\hat{C}_a a - C_p a_p < \epsilon$ then return \hfill \begin{itemize} \item \texttt{If signal coding did not change significantly, stop} \end{itemize}
9: else
10: \(a_p = \hat{a}, \ C_p = \hat{C}_a \)
11: end if
12: end while
**Output:** $\hat{a}, \hat{C}_a$

5. Experimental results on signal restoration

Next, we have evaluated the effectiveness of our model for various image restoration tasks on both synthetic and real data. In signal restoration, a high quality signal $x$ needs to be reconstructed from its degraded measurements $y$. The problem can be modeled in a generic form as

$$y = Hx + v$$

where $H$ is a degrading operator and $v$ is additive noise.
5.1. Synthetic data

For the case of synthetic data, we have used a dictionary of 300 gaussian anisotropic atoms with mother function $\phi(x,y) = A \exp(-((x/2)^2 - y^2))$. We have sampled the image plane for two scale levels $[0.5 \ 1]$ with a step size 1 for translation and $\pi/6$ for rotation. The atoms of the dictionary were combined according to 10 predefined molecules contained in $C_s$. The size of the signals and the molecules was $10 \times 10$. Each molecule was randomly constructed to contain 2, 3 or 4 atoms of equal energy. Then each signal was created as a random combination of a few molecule realizations (2, 3 or 4).

To produce a molecule realization $c_{i,j}$ of a molecule prototype $c_{s,j}$ we use the following procedure. We start by identifying the atoms that participate in the molecule prototype i.e., the support $\Gamma_{i,j}$. Then, for each atom $d_i \in \Gamma_{i,j}$ we produce an approximation using the atoms in the atom pool $P(d_i)$. More specifically, we first decide on the number $k$ of atoms from $P(d_i)$ that we will use in the place of $d_i$ by sampling a geometric distribution with $p = 0.7$. In this way, we make sure that the approximation is a sparse combination of a few atoms. Then, we randomly pick $k$ atoms from $P(d_i)$, denoted as $P_k(d_i)$, and we use for each atom $d_k \in P_k(d_i)$ we assign a coefficient $c_{s,j}(n)$ so that the projection of the combination $\sum_{n \in P_k(d_i)} c_{s,j}(n)d_n$ to the direction of $d_i$ is close to the original coefficient value $c_{s,j}(n)$, i.e., $0.95 \leq \sum_{n \in P_k(d_i)} c_{s,j}(n)\langle d_n, d_i \rangle \leq 1.05 c_{s,j}(n)$.

We have compared our method with the $l_1-2$ group norm (the algorithm is denoted as $A_{12}$) [10]. To define each group $g_i \in G$ we use the support of the corresponding molecule $m_i$. The atoms that did not belong to any group, were considered as separate groups of size 1. The resulting optimization problem was:

$$\hat{b} = \arg \min_b \left\{ ||y - H D b||_2 + \lambda \sum_{k \neq g} ||b_k|| \right\} \quad (18)$$

where $b$ is the signal decomposition in the atomic level and $b_k$ is its restriction on $g_k$. The decomposition $\hat{a}$ in groups (or equivalently molecules in our case) is computed as the $l_2$ norm of the coefficients in each group i.e., $\hat{a}_i = ||\hat{b}_i||$.

As we have discussed before in Section 3, one alternative for the synthetic dictionary is the dictionary of molecules prototypes. This approach is similar to the sparse coding step in [13]. We have also compared our scheme with sparse coding with $l_1$ regularization on the molecule dictionary (we denote this algorithm as $A_m$), i.e., the outcome of:

$$\hat{a} = \arg \min_a \left\{ ||y - H D_a * a||_2 + \lambda ||a||_1 \right\} \quad (19)$$

where $D_a = DC_s$ is the molecule dictionary.

Finally, we have also compared our scheme with simple sparse coding on $D$, i.e.,

$$\hat{a} = \arg \min_a \left\{ ||y - H * D * a||_2 + \lambda ||a||_1 \right\} \quad (20)$$

The method is denoted $A_0$ in the rest.

The performance of the algorithms is compared using various measures. To quantify the performance in terms of the signal recovery we have computed both the mean square reconstruction error of the signal approximation (MSRE), i.e., $\sum_{x=0}^{|x|} \frac{x^2 - x^2}{\sum_{x=0}^{|x|}}$ where $x$ is the signal reconstruction and $N$ is the number of signals, as well as the mean sparsity ratio of the recovered representations where the sparsity ratio is computed as the $l_0$ norm of the recovered representation in $D$ over the $l_0$ norm of the true atomic representation. Finally, we are also interested in how effective are the schemes in detecting the correct molecules. Therefore, we have also computed the accuracy of the molecule detection, which is the ration of the correctly categorized molecules $(TP + TN)$ over all the molecule instances $(P + N)$.

5.1.1. Denoising

To start with, we have tested the performance of the schemes under noise. In this case, $H = I$ and $a$ is white gaussian noise. The results, for different noise levels, are shown in Fig. 7. For each noise level, the results were averaged over 5 different molecule matrices and 1000 signal instances per matrix. The parameters for each algorithm, chosen based on a small validation set, were: $\lambda_1 = 0.01$, $\lambda_2 = 1$, $\lambda_3 = 0.1$ for AMC and $\lambda = 0.1$ for all the rest. From Fig. 7 we can observe that as the noise increases the effectiveness of the structure is more prominent: the MSRE of $A_0$ progressively deteriorates compared to the other 3 schemes that use a structured prior. Moreover, for the highest noise level the $A_m$ scheme which is the one with the least flexible structure prior, almost reaches the best performance. However, our scheme manages to perform best for all the noise levels by uncovering signal representations with small MSREs, accurate molecule detection, and satisfactory sparsity ($A_m$ has a fixed sparsity level for each molecule, therefore it is expected to have the lower value as the most constrained one).

5.1.2. Inpainting

Next, we have tested the performance of the schemes for inpainting. In this case, we have created a set of signals by omitting the signal values in a randomly chosen square region. We have tried three different sizes for the region: $3 \times 3$, $4 \times 4$ and $5 \times 5$. Then, the signals were divided into 4 sets based on their SNR. The signal recovery problem was solved over the known regions of the signals: each signal $x$ was expressed as $x' = P_5 * x$ where $P_5$ is the mask denoting the known region. In this case, $H = P_5 * I$ resulting in masking each dictionary atom. No extra noise was

Fig. 7. The results for denoising on synthetic data with different coding schemes. The performance is evaluated with the MSRE of the reconstructed signals as well as the sparsity ratio and the accuracy of the recovered representations.
added to the data. The values for the parameters were $\lambda_1 = 0.001$, $\lambda_2 = 1$, $\lambda_3 = 0.1$ for AMC and $\lambda_1 = 0.01$ for all the rest. The results are shown in Fig. 8. Again, we can observe the benefits from the flexible prior that our scheme provides compared to the rest: the MSRE is always the lowest, the accuracy is the highest while the sparsity ratio is satisfactory, usually the lowest after $A_m$ which is the most constrained one. In case of highly disturbed signals (lowest SNR) the $A_m$ also outperforms the rest, proving the importance of structure in applications were there is a significant amount of missing information.

5.1.3. Compressed sensing

Finally, we have compared the recovery performance of the schemes for compressed sensing. The measurement process was performed by setting $H = \Phi$ where $\Phi$ is a random projection matrix. The entries of $\Phi$ were independent realizations from a standard normal distribution. We have checked three different sizes for $\Phi$ namely 25, 15 and 8 measurements. For each number of measurements the results were averaged over 5 different instances of matrix $\Phi$. The values of the parameters were $\lambda_1 = 0.01$, $\lambda_2 = 10$, $\lambda_3 = 0.01$ for AMC and $A_1$ while $\lambda_1 = 1$ for $A_m$ and $\lambda_1 = 0.01$ for $A_{12}$. The results for the different number of measurement are shown in Fig. 9. Our scheme significantly outperforms the rest as the number of measurements decreases while keeping a high accuracy on molecule detection. The sparsity ratio is almost stable for all sizes of measurement matrix and quite close to 1 which is the desired value.

5.2. Handwritten digit images

Next, we have used our adaptive molecule coding scheme to perform denoising on MNIST images [29]. The images have been downsamled to $14 \times 14$ and normalized. In order to better fit the signal model the digits were further coarsely pre-aligned to avoid big discrepancies in the position and the orientation. The molecule prototypes were extracted using the algorithm presented in [13] from 1000 examples per digit while for the testing we used 100 examples per digit. The denoising performance was tested over different noise levels and measured by the mean squared reconstruction error and the mean sparsity ratio. The parameters were fixed according to a small validation set and their values were $\lambda_1 = 0.001$, $\lambda_2 = 0.01$, $\lambda_3 = 0.01$ for AMC and $\lambda_1 = 0.01$ for the rest of the schemes.

The results of our experiments are presented in Fig. 10. We have experimented with both denoising each digit separately using molecules extracted only for its class as well as denoising with molecules extracted from many classes simultaneously. In the first two columns we show the results we obtained for digits 0 and 9 separately while in the third column we plot the results for the case of denoising digits 0, 1, 2 and 3 with molecules extracted for all 4 digits together. From the plots we can see that AMC is the scheme that manages to perform well for all different noise levels. As expected the benefits from rich structure priors are more prominent in the presence of severe noise, where $A_m$, the scheme with the most restrictive prior, outperforms $A_1$ and $A_{12}$ that have loser priors. However, for lower noise levels the performance of $A_m$ is not sufficiently good due to the rigidity of its prior. Our scheme on the other hand performs well in all cases as it adapts to the signals almost as successfully as $A_1$ in the less noisy cases, while preserving the structure as $A_m$ in the more noisy cases. Finally, it is also important to note that AMC is the scheme that achieves on average a sparsity ratio close to one, meaning that it is highly efficient as it achieves a good signal restoration using only as many components as it is necessary.
5.3. Natural images

Finally, in image restoration it is often the case that the non-local similarities in different regions of images are used to enhance the restoration process [30,31]. The idea of 'nonlocally centralized' sparse codes is not very far from the idea of molecule prototypes. Therefore, we have followed the same intuition to define molecules prototypes based on the non-local similarity of patches and use their deformed versions to further enhance the image recovery.

To be more specific, when only sparsity is used as a prior for the recovery of the patches $x_i$ of an image $X$, the recovery problem for each patch can be written as:

$$\hat{a}_i = \arg \min_{a_i} \| y_i - HDa_i \|_2^2 + \lambda_1 \| a_i \|_1$$

(21)

where $a_i$ is the decomposition of the patch $x_i$ in the dictionary $D$ and $y_i$ is the vector of measurements acquired for this patch i.e.,

$$y_i = Hx_i + v$$

(22)

with $v$ potential additive noise. The recovered image $\hat{X}$ is then created by the recovered patches $\hat{x}_i$.

However, when taking into account the non-local similarity of the patches, a molecule prototype can be extracted for each patch and further enhance the recovery by restricting the code of the each patch to be a realization of the prototype. The corresponding coding problem is then:

$$\tilde{c}_{ij} = \arg \min_{c_{ij}} \| y_{ij} - H D c_{ij} \|_2^2 + \lambda_2 \| W_i \times (c_{ij} - S \ast c_{ij}) \|_2^2 + \lambda_3 \| c_{ij} \|_1$$

(23)

where $c_{ij}$ is the molecule prototype for $x_i$ and $c_{ij}$ is the patch dependent molecule realization. In order to obtain $c_{ij}$ we search the image $X$ for the most similar patches to $x_i$ and we build a set $\Omega_i$ as in [30]. Then, based on the sparse codes of the patches in $\Omega_i$ we extract a molecule prototype for $\hat{x}_i$. The prototype extraction algorithm is a greedy procedure that identifies a small number of atoms to account for most of the energy in the sparse codes in $\Omega_i$ while taking into account the atoms pools. It is an iterative procedure that at each step adds in the support of the molecule prototype the atom with the most energy in its pool. The energy of the atoms falling in the already chosen pools is considered covered and the algorithm iterates until a sufficient amount of the energy is covered. In this way, we extract a molecule prototype $c_{ij}$ that accepts as realizations all the patches in $\Omega_i$.

To show that our proposed coding scheme is suitable for enhancing the recovery of the original image, we have compared it to the $\lambda_1$ based sparse coding presented in Eq. (21) which only imposes sparsity as structure. Moreover, following the ideas in [30], we have also implemented a scheme where the imposed structure is defined as the mean sparse code over similar patches. The corresponding optimization problem is then:

$$\tilde{a}_i = \arg \min_{a_i} \| y_i - HDa_i \|_2^2 + \lambda_2 \| \tilde{a}_i - a_i \|_2^2 + \lambda_3 \| a_i \|_1$$

(24)

where $\tilde{a}_i$ is the mean sparse code obtained from the sparse codes of the patches in $\Omega_i$. In the plots, this scheme is denoted as ‘Mean’.

In the rest, we present the results that we obtained for the case of denoising, compressed sensing and inpainting for the natural images ‘House’ and ‘Barbara’. In all cases, the images were divided in $10 \times 10$, non-overlapping patches. As a base dictionary $D$ we have used a DCT overcomplete dictionary with 256 atoms. For solving the coding problem in Eq. (23) we have used Algorithm 1, namely the part that solves for $C_k$ given $a$, as in this case for each patch there is only one molecule prototype and as result the vector of molecule coefficients is set to 1.
5.3.1. Denoising

To start with, we have tested the performance of the different schemes in noisy settings. In this case, $H = I$ and $v$ is white Gaussian noise in Eq. (22). The PSNR of the recovered images for different noise levels is shown in Fig. 11. For each noise level, the results are averaged over 5 different instances of noisy images. We can observe that the non-local similarity of the patches is very effective for the denoising if combined with our molecule based coding scheme as both the $l_1$ sparse coding and the mean prior from Eq. (24) result in a much lower PSNR performance than our proposed structured sparsity solution. This can be explained by the fact that the $l_1$ does not take into account all the similarities between the patches. On the other hand, the ‘Mean’ prior takes into account the similarities but it constrains the sparse representation of every patch to lie close to the average sparse code from similar patches and as a result it can easily smooth out details in the images.

5.3.2. Compressed sensing

For the compressed sensing, the operator $H$ was set equal to $\Phi$, a random projection matrix whose entries are independent realizations from a standard normal distribution. We have checked two different sizes for $\Phi$, namely 30 and 50 measurements, and for each number of measurements the results have been averaged over 5 different instances of the matrix $\Phi$. The measurements were further corrupted with Gaussian noise. In Fig. 12 we show the PSNR of the recovered images based on the three different sparsity-based...
outperform the results in Fig. 13, the benefit from taking into account the similarity of the patches is very helpful for the image restoration as the $l_1$ sparse coding has a much lower PSNR than the other two schemes. Moreover, our molecule-based coding scheme manages to extract more effectively the structural similarities of the patches than the mean sparse code as it achieves better PSNR results for the majority of settings. Therefore it confirms that the idea of molecule prototypes and realizations based on atoms pools is a powerful model that provides correct priors for patch based restoration of images.

5.3.3. Inpainting

Finally, we have compared the different restoration schemes in the problem of inpainting. In this case, we have corrupted the images by omitting the signal values in randomly chosen pixels. We have tried three different percentages of missing pixels, namely 50%, 60% and 70%. The image recovery problem is solved over the known regions of the patches: each patch $x_i$ is expressed as $x_i = P_i \star x$ where $P_i$ is the mask denoting the known region. In this case, $H = P_i \star I$ in Eq. (22) resulting in masking each dictionary atom. No extra noise is added to the data. As we can observe from the results in Fig. 13, the benefit from taking into account the similarities of the patches is prominent as both AMC and the ‘Mean’ outperform the $l_1$ sparse coding. In this case, the performance of AMC is similar to the performance the ‘Mean’ prior which means that the extracted prototypes in AMC have not captured successfully the structure details of the patches. This is reasonable as a big percentage of the pixels, i.e., a lot of details, is missing. For such restoration cases, it is necessary to adopt a different technique for extracting the prototypes that would be based on a more elaborate learning scheme.

6. Conclusions

In this paper we have presented a new two-layer structure model for signals. We have defined our structural elements, the molecules, as linear combinations of atoms and we have distinguished between molecule prototypes and molecule realizations based on the notion of pools of atoms. The addition of coefficients in the structure permits a better modeling of higher level patterns while the definition of molecule realizations results in extra invariance to small deformations of patterns. We have presented our new algorithmic scheme for adaptive molecule coding (AMC) and we have conducted experiments on both synthetic and real data that proved the effectiveness of our model for various restoration tasks.

Acknowledgments

We would like to thank Prof. Christine Guillemot for the fruitful discussions and comments. This work was been partially funded by the Swiss National Science Foundation (SNSF) under Grant 200021_143999.

Appendix A. Bound on error of atom realization

As we have mentioned in Section 2.2, if we constrain the atoms that participate in the realization of the atom $d_i$ to lie in its pool $P(d_i)$ and have non-negative coefficients we can guarantee that the resulting approximation has a bounded error, i.e., $||d_i - v_i||_2 \leq L$. To see why, let $v_i = \sum_{j \in P(d_i)} b_j d_j$. Then, from Fig. A.14 we have:

$$||d_i - v_i||_2^2 = ||r_i||^2 = ||p_i||^2 + (1 - v_i)^2$$

$$= e_i^2 \tan^2 \phi_{vi} + (1 - e_i)^2$$

(A.1)

However for the angle between $v_i$ and $d_i$ we have:

$$\cos \phi_{vi} = \frac{\langle v_i, d_i \rangle}{\| v_i \|} = \frac{\sum_{j \in P(d_i)} b_j \langle d_i, d_j \rangle}{\| \sum_{j \in P(d_i)} b_j d_j \|} \geq \frac{(1 - \epsilon) \sum_{j \in P(d_i)} b_j}{\sum_{j \in P(d_i)} |b_j|} = 1 - \epsilon$$

if $b_j \geq 0, \forall j \in P(d_i)$. Therefore, when we allow only non-negative coefficients in the approximation, $v_i$ belongs in $P(d_i)$.

Fig. A.14. An example of the realization of the atom $d_i$ from vector $v_i = b_1 d_1 + b_2 d_2 + b_3 d_3$ with $d_1, d_2, d_3 \in P(d_i)$ and $b_1, b_2, b_3 > 0$. 

![Fig. 13. Results for image recovery in case of inpainting. On the vertical axis we have the PSNR of the recovered image and on the horizontal the percentage of missing pixels in the image. The values of the parameters for this application were set to $\lambda_1 = \lambda_2 = \lambda_3 = 0.1$.](image-url)
Moreover, since \( \cos \phi_k > 1 - \epsilon \), then \( \sin \phi_k \leq \sqrt{\epsilon(1-\epsilon)} \) and therefore \( \tan \phi_k \leq \epsilon \). Finally, from Eq. (A.1) we get:

\[
\|d_i - v_i\|^2 \leq (1 - \epsilon)^2 + \epsilon^2 \frac{\epsilon}{(1 - \epsilon)} \tag{A.2}
\]

**Appendix B. Recovery analysis supplementary material**

We now present the theorems that provide the lower and upper bounds on the coherence of dictionaries \( D_{C_k} \) and \( D_{C_0} \) discussed in Section 3. The dictionary \( D_{C_k} \) is a dictionary that contains more than one realizations per molecule prototype while the dictionary \( D_{C_0} \) is restricted to one realization per prototype. To evaluate their coherences denoted as \( \mu_k \) and \( \mu_0 \) respectively we will first need to examine the distance between a molecule prototype \( m_{x,i} = D_{C_{x,i}} \) and its possible realizations \( m_{x,i} = D_{C_{x,j}} \). The corresponding upper bound is presented in the next theorem.

**Theorem 1.** Let \( \|c_{x,i}\|_0 \leq n, \forall i \) and \( \phi = \cos(1 - \epsilon) \) where \( \epsilon \) is the parameter used in the pool definition in Eq. (4). Moreover, let the error \( |c_{x,i}(i) - e_i| \) between the energy in an atom \( d_i \), of a molecule prototype and the energy on its pool in any of the realizations be bounded by \( |c_{x,i}(i) - e_i| \leq E C_{x,i}(i), \forall i \in \Gamma_{x,i} \), where \( E \) is a positive constant. Finally, let \( \mu_{\delta \phi} \) stand for the in-molecule coherence defined as the maximum coherence between the atoms that belong to the same molecule, i.e., \( \mu_{\delta \phi} = \max \left( \max_{j \neq i} \frac{|c_{x,i}(j) - c_{x,j}(i)|}{d_i, d_j > 1} \right) \) and assume that \( \mu_{\delta \phi} \leq \frac{1}{1 - \epsilon} \). Then, the distance between any molecule prototype \( m_{x,i} \), and any of its realizations \( m_{x,j} \), is bounded by

\[
\|m_{x,i} - m_{x,j}\| \leq \sqrt{\frac{(1 + E^2)\tan^2 \phi + E^2}{1 - (n - 1)\mu_{\delta \phi}}} \tag{B.1}
\]

**Proof.** For the molecule prototype \( m_{x,i} = \sum_{i \in \Gamma_{x,i}} c_{x,i}(i)d_i \), a molecule realization can be written as:

\[
m_{x,j} = \sum_{i \in \Gamma_{x,i}} v_i = \sum_{i \in \Gamma_{x,i}} (e_i d_i + p_i) = m_{x,i} + \sum_{i \in \Gamma_{x,i}} (p_i - |c_{x,i}(i) - e_i|d_i)
\]

where an example of an approximation vector \( v_i \) for an atom \( d_i \) is shown in Fig. B.15. Therefore:

\[
\|m_{x,j} - m_{x,i}\| = \sum_{i \in \Gamma_{x,i}} (p_i - |c_{x,i}(i) - e_i|d_i) \leq \sum_{i \in \Gamma_{x,i}} |p_i - (c_{x,i}(i) - e_i)|d_i \tag{B.1}
\]

by the triangle inequality. However, \( p_i \) is orthogonal to the direction of \( d_i \). Therefore:

\[
\|p_i - (c_{x,i}(i) - e_i)d_i\| = \sqrt{(p_i)^2 + (|c_{x,i}(i) - e_i|d_i)^2} = \sqrt{\epsilon^2 \tan^2 \phi + (|c_{x,i}(i) - e_i|)^2}
\]

Substituting in Eq. (B.1), we get:

\[
\|m_{x,j} - m_{x,i}\| = \sum_{i \in \Gamma_{x,i}} \sqrt{\epsilon^2 \tan^2 \phi + (|c_{x,i}(i) - e_i|)^2} \leq \sqrt{(1 + E^2)\tan^2 \phi + E^2}\|c_{x,i}\| \tag{B.2}
\]

since \( |e_i| \leq EC_{x,i}(i), \forall i \in \Gamma_{x,i} \), and \( c_{x,i}(i) \geq 0, \forall i,i \). For the \( \|c_{x,i}\|_0 \), we have:

\[
\|c_{x,i}\|_0 \leq \sqrt{n} \tag{B.3}
\]

To bound the \( l_2 \) norm, we use the Rayleigh quotient \( R(M,x) = \frac{x^TMx}{x^Tx} \) and its bound \( \lambda_{\text{min}}(M) \leq R(M,x) \). In our case, \( M = D^T_{x,i}D_{x,i} \) where \( D_{x,i} \) is the matrix of the atoms participating in molecule \( m_{x,i} \). Then, for \( x = m_{x,i} \) we have:

\[
\lambda_{\text{min}}(D^T_{x,i}D_{x,i}) \leq \frac{1}{\|c_{x,i}\|_2} \Leftrightarrow \|c_{x,i}\|_2 \leq \frac{1}{\sqrt{\lambda_{\text{min}}(D^T_{x,i}D_{x,i})}} \tag{B.4}
\]

where \( \lambda_{\text{min}} \) is the minimum eigenvalue of \( D^T_{x,i}D_{x,i} \). Finally, from the Gershgorin circle theorem applied on \( D^T_{x,i}D_{x,i} \), which is the Gram matrix of \( D_{x,i} \) that contains the inner products of the atoms in \( \Gamma_{x,i} \), we get:

\[
|\lambda - 1| \leq \max_{j \neq i} \sum_{j \in \Gamma_{x,j}} |d_i, d_j > | \leq \mu_{\delta \phi} \leq \frac{1}{1 - \epsilon} \tag{B.5}
\]

Since \( \mu_{\delta \phi} = \max \left( \max_{i,j} \frac{|c_{x,i}(j) - c_{x,j}(i)|}{d_i, d_j > 1} \right) \) we get that \( \forall i,j \):

\[
1 - (n - 1)\mu_{\delta \phi} \leq \lambda_{\text{min}}(D^T_{x,i}D_{x,i}) \tag{B.5}
\]

Assuming \( 1 - (n - 1)\mu_{\delta \phi} > 0 \Leftrightarrow \mu_{\delta \phi} \leq \frac{1}{1 - (n - 1)\mu_{\delta \phi}} \), and substituting in Eq. (B.4), we have:

\[
\|c_{x,i}\| \leq \frac{1}{\sqrt{1 - (n - 1)\mu_{\delta \phi}}} \tag{B.5}
\]

Combining Eqs. (B.3), (B.5) and (B.2) we finally get that:

\[
\|m_{x,j} - m_{x,i}\| \leq \sqrt{\frac{(1 + E^2)\tan^2 \phi + E^2}{1 - (n - 1)\mu_{\delta \phi}}} \tag{B.6}
\]

With an established bound for the distance \( \|m_{x,j} - m_{x,i}\| \) between a molecule prototype and its realizations, we can prove the following theorem which provides a lower bound for the coherence \( \mu_k \) of any dictionary \( D_{C_k} \) with more than one realizations per prototype.

**Theorem 2.** When the distance between any molecule prototype and its realizations is bounded by \( \|m_{x,j} - m_{x,i}\| \leq \tau \) with \( \tau < \frac{1}{2\epsilon} \), the coherence \( \mu_k \) of any dictionary \( D_{C_k} \) with more than one molecule realizations per molecule is

\[
\mu_k \geq 1 - 2\tau^2 = L_k \tag{B.6}
\]

**Proof.** The coherence of the dictionary \( D_{C_k} \) is:

\[
\mu_k = \max_{x,y,k} \frac{|m_{x,i} - m_{y,j}|}{\|m_{x,i}\| \|m_{y,j}\| \cos \phi_{m_{x,i},m_{y,j}}} = \max_{x,y,k} \|m_{x,i}\| \|m_{y,j}\| \cos \phi_{m_{x,i},m_{y,j}} \]

\[
\frac{1}{\sqrt{1 - (n - 1)\mu_{\delta \phi}}} \tag{B.5}
\]

![Fig. B.15.](image-url) An example of the approximation of the atom \( d_i \) from vector \( v_i \), deviating by \( \phi \), in direction. The desired energy level is \( c_i \) while the projection of \( v_i \) gives an energy of \( e_i \).
where \( m_{1,l} \) and \( m_{2,l} \) are realizations of the molecule prototypes \( m_{1} \) and \( m_{2} \), and \( \phi_{m_{1,l} m_{2,l}} \) is the angle between the two vectors. A lower bound to \( \mu \) can be found by computing the maximum angle between two realizations of the same molecule, i.e. for \( l = k \). Then, \( \mu_k \geq \max \{ \cos \phi_{m_{1,l} m_{2,l}} \} \) \( \forall l \).

From Fig. B.16 we can see that, since all the molecule realizations live in a sphere of radius \( r \) around the prototype \( m_{1,l} \), the angle between any two realizations \( m_{1,l}, m_{2,l} \) has to be less than or equal to \( 2 \phi_k \). For the bound to be different than zero, we need that \( 2 \phi_k < \pi/2 \iff r < \sqrt{2}/2 \). Then, from Fig. B.16, we have:

\[
\cos \phi_k = \frac{||OC||}{||Om_2||} = \sqrt{1 - r^2} = \sqrt{1 - r^2}
\]

since \( ||Om_2|| = \|m_{1,l}\| = 1 \). Therefore:

\[
\phi_{m_{1,l} m_{2,l}} \geq 2 \phi_k \iff \cos \phi_{m_{1,l} m_{2,l}} \geq \cos 2 \phi_k \iff \phi_{m_{1,l} m_{2,l}} \leq \frac{\pi}{4} \iff \cos \phi_{m_{1,l} m_{2,l}} \leq 2 \cos^2 \phi_k - 1, \quad \phi \leq \frac{\pi}{4} \iff \cos \phi_{m_{1,l} m_{2,l}} \geq 2(1 - r^2), \quad r < \sqrt{2}/2 \iff \mu_k \geq 1 - 2r^2, \quad r < \sqrt{2}/2 \quad (B.7)
\]

Finally, we can use the same bound on the distance \( \|m_{1,l} - m_{2,l}\| \) between a molecule prototype and its realizations to establish an upper bound on the coherence \( \mu_k \) of any dictionary \( DC_u \) with maximum one realization per prototype. The following theorem formalizes this bound.

**Theorem 3.** Let the coherence of the molecule prototype dictionary \( DC \) be \( \mu \). Given the bound on the distance between any molecule prototype and its realizations \( \|m_{1,l} - m_{2,l}\| \leq \mu \) with at most one realization per molecule is

\[
\mu_k \leq U_k = \mu(1 - 2r^2) + 2r \sqrt{(1 - \mu^2)(1 - r^2)}
\]

**Proof.** We have:

\[
\mu_k = \max_{y, \|k \| = k} \frac{\|m_{1,l} m_{2,l}\|}{\|m_{2,l}\|} = \max_{x \|k \| = k} \cos \phi_{m_{1,l} m_{2,l}}
\]

where \( m_{1,l}, m_{2,l} \) are realizations of the molecules \( m_{1} \) and \( m_{2} \) respectively and \( \phi_{m_{1,l} m_{2,l}} \) is the angle between the two corresponding vectors. In the opposite case, a similar analysis can be followed and the final bound on \( \mu_k \) is the same. Under this assumption, we have

\[
\mu_k = \max_{y, \|k \| = k} \cos \phi_{m_{1,l} m_{2,l}}
\]

Moreover, we can assume that the indices \( l, k \) that maximize \( \|m_{1,l} - m_{2,l}\| \) are the same as the ones that maximize the equation \( \mu = \max_k \|m_{1,l} - m_{2,l}\| = \max_k \cos \phi_{m_{1,l} m_{2,l}} \). In other words, we assume that the molecule prototypes that are the most coherent are also the ones that give rise to the most coherent realizations. Therefore, we will continue our analysis for the case where \( \cos \phi_k = \mu \). It is sufficient to restrict the rest of the analysis on the plane defined by the molecules prototypes \( m_{1,l}, m_{2,l} \). This is true because the space occupied by each prototype’s realizations is a sphere, and the minimum distance and angle points between spheres live on the plane defined by their centers.

The geometry on this plane is shown in Fig. B.16. From the figure we have that:

\[
\phi_{m_{1,l} m_{2,l}} \geq \phi \quad \text{and} \quad \phi = \phi_k - 2 \phi_k
\]

Therefore:

\[
\phi_{m_{1,l} m_{2,l}} \geq \phi_k - 2 \phi_k \iff \cos(\phi_k - 2 \phi_k) \geq \cos \phi_{m_{1,l} m_{2,l}}
\]

Moreover, using Eq. (B.10), we have:

\[
\mu_k \leq \cos(\phi_k - 2 \phi_k)
\]

However, from trigonometry we have:

\[
\cos(\phi_k - 2 \phi_k) = \cos \phi_k \cos 2 \phi_k + \sin \phi_k \sin 2 \phi_k
\]

Since \( \cos \phi_k = \mu \), we also have \( \sin \phi_k = \sqrt{1 - \cos^2 \phi_k} = \sqrt{1 - \mu^2} \). Moreover from the triangle \( OCm_{1,l} \) we have \( \cos 2 \phi_k = 1 - 2r^2 \) and \( \sin 2 \phi_k = \sqrt{1 - \cos^2 2 \phi_k} = \sqrt{1 - (1 - 2r^2)^2} = 2r \sqrt{1 - r^2} \). Substituting the above in Eq. (B.13) we get:

\[
\cos(\phi_k - 2 \phi_k) = (1 - 2r^2) + 2r \sqrt{(1 - \mu^2)(1 - r^2)}
\]

Substituting this expression in Eq. (B.12), we get:

\[
\mu_k \leq \mu(1 - 2r^2) + 2r \sqrt{(1 - \mu^2)(1 - r^2)}
\]

**References**


