Multiscale Adaptive Representation of Signals

Cheng Tai

Abstract

Starting with dictionary learning, we want to address two issues that we consider important. One is the inefficiency of sparse coding used by most dictionary learning scheme to encode a new incoming signal; the other is the lack of a truly multi-scale representation.

1 Overview of dictionary learning and wavelets

It is now well acknowledged that sparse and redundant representations of data plays a key role in many signal processing areas. The ability to represent a signal as a sparse linear combination of a few atoms from a possibly redundant dictionary lies in the heart of many models in signal processing, such as image/audio compression, denoising, and some higher level tasks such as pattern recognition.

Over the years, many efforts have been put on designing dictionary with certain properties. There are two lines towards this objective. One line can be loosely called the analytic dictionary, which assume the mathematical properties of the class of signals of interest, and develops optimal representations for that class of signals. Examples include: Fourier basis, wavelets, wavelet tight frames, curvelets, contourlets, etc. The second kind takes a different route, it doesn't make assumptions about the mathematical properties of the class of signals, and tries to learn the dictionaries from the data from scratch. Beginning with the seminal work of Olshausen and Field, the field of dictionary learning has seen many promising advances.

Models of the first kind are characterized by mathematical formulation and implicit transformations whereas the advantage of models of the second kind is that they lead to state-of-the art results in many practical low level signal processing applications.

Despite the elegance and success of this methodology, there are also some important short comings to it, these include:

- Computational cost is high. For the signals of length N, the trained dictionary $A \in \mathbb{R}^{m\times N}$ is stored and used explicitly. Each multiplication of the form Ax or $A^T y$ requires $O(mN)$ operations. In comparison, the analytic dictionary approach, such as Fourier transform only takes $O(N \log N)$ operations, and one level of wavelet transform take only $O(N)$ operations. When plugged to applications, we have to solve a relatively complex sparse coding program, even in the fastest greedy algorithms, such as Matching Pursuit, the matrix multiplications are carried out several times. Thus, because the learned dictionary is unstructured, it is much less efficient compared to traditional transform methods.
- Restriction to low-dimensions. Because the learning procedure requires solving a relative complex non-convex optimization program, the signals that can be practically trained is restricted to low dimensions, typically, $n \leq 1000$ is a reasonable limit. That is also a reason that in image processing applications, most popular models only train dictionaries on small image patches. An attempt to go beyond the limit raises a series of problems, the need for an huge amount of training data and intolerable training time.
- Operating on a single scale. Dictionaries as obtained by the MOD and the K-SVD operate on signals at a single small scale. Past experience with wavelets has taught us that often times it is beneficial to process the signals at several scales, and operates on each scale differently. This is actually related to the previous shortcomings, since the dictionary atoms that can be trained is small in size, which does not allow much room for multiple scales. There are some attempts in training multi-scale dictionaries, but this direction has not been thoroughly explored.
- Artifacts. As the dictionary operates on image patches, in tasks such as image compression, the patch wise operation produces visually unpleasant block effects along the boarders of the patch. Post processing is often needed to remove these artifacts.

A natural question is, can we devise a way to get the best of both worlds? It is the goal of this paper to propose a partial solution to this question. In particular, we want to address two issues: the inefficiency of sparse coding used by most dictionary learning scheme to encode a new incoming signal and lack of a truly multi-scale representation. We will devise a novel representation of image signals with some favorable properties, such as multi-scale, computationally efficient, and is adapted to the signals as dictionary learning does.

2 A First Attempt to Improve Computational Efficiency of Dictioanry Learning

There are two routes we can take to reach our goal. One starts from dictionary and the other from wavelet tight frames. A route which we will not detail but will ultimately lead us to the same goal is the following: start with image patches, instead of training unstructured dictionary, we train tight frames, this helps improving the computational efficiency in that we don't need to solve a sparse coding program any more, instead, just perform one matrix vector multiplications. Next, we make the dictionary convolutional, based on the premise that image patches are shift invariant at a

certain scale. Then, we would reach at something similar to a one layer adaptive wavelet tight frame transform.

3 Single Scale Adaptive Wavelet Tight Frames

Another line parallel to dictionary learning and also has seen success in image processign tasks is the applications of wavelets. The traditional wavelets are universal basis. A key factor to the success of wavelets is that it is assumed and experimentally tested on larges datasets that natural images often have a sparse representation under the wavelet basis. Hence, although not as explicitly pursued as in dictionary, wavelets also shares sparsity property, which we think is key to a good representation of imags.

Recognizing the importance of sparsity, we shall continue pursuing it in our constructions of new representations. On the other hand, the superiority of dictionary learning based approaches over wavelet shrinkage shows the importance of adaptivity. Hence we keep in mind that a good representation of images should have at least three qualities: sparsity, computational efficiency, and should be multi-scale. And this is a challenge.

The plan is as follows: we first make the wavelet adaptive to the signal class at hand, hence improving sparsity while keeping its computational efficiency, then we extend the representation to multiple scales.

3.1 Short Introduction To Wavelet Tight Frames

In this subsection, we give a very brief introduction to wavelet tight frames, which is necessary for our later construction. For detailed survey, the readers may refer to []. Let $\mathcal{H}$ be a Hilbert space, a sequence $\{x_n\} \subset \mathcal{H}$ is called a tight frame for $\mathcal{H}$ if

$$\|x\|_2^2 = \sum_n |\langle x, x_n\rangle|^2, \quad \text{for any } x \in \mathcal{H}$$

Associated with a tight frame are two operators, one is the analysis operator defined by

$$W : x \in \mathcal{H} \to \{\langle x, x_n\rangle\} \in l_2(\mathbb{N})$$

and the other is its adjoint operator W^T called the synthesis operator:

$$W^T : \{a_n\} \in l_2(\mathbb{N}) \to \sum_n a_n x_n \in \mathcal{H}.$$

The sequence $\{x_n\} \subset \mathcal{H}$ is called a tight frame if and only if $W^T W = I$, where $I : \mathcal{H} \to \mathcal{H}$ is the identity operator. In other words, given a tight frame $\{x_n\}$, we have the following canonical expansion:

$$x = \sum_n \langle x, x_n\rangle x_n, \quad \text{for any } x \in \mathcal{H}$$

The sequence $Wx := \{\langle x, x_n, \rangle\}$ is called the canonical tight frame coefficient sequence. Thus, tight frames are often viewed as generalizations of orthonormal basis. In fact, a tight frame $\{x_n\}$ is an orthonormal basis for $\mathcal{H}$ if and only if $\|x_n\| = 1$ for all x_n.

One widely used class of tight frames in image processing is the discrete wavelet tight frames generated by a set of filters $\{a_i\}_{i=1}^m$. Given a filter $a \in l_2(\mathbb{Z})$, define the linear convolution operator $S_a : l_2(\mathbb{Z}) \to l_2(\mathbb{Z})$ by

$$[S_a(v)](n) := [a * v](n) = \sum_{k\in\mathbb{Z}} a(n-k)v(k), \forall v \in l_2(\mathbb{Z})$$

For a set of filters $\{a_i\}_{i=1}^m \subset l_2(\mathbb{Z})$, we define its analysis operator W by

$$W = [S_{a_1(-\cdot)}, S_{a_2(-\cdot)}, \cdots, S_{a_m(-\cdot)}]^T.$$

Its synthesis operator is defined as the transpose of W:

$$W^T = [S_{a_1}, S_{a_2}, \cdots, S_{a_m}].$$

The rows of W form a tight frame if and only if $W^T W = I$. The Unitary Extension Principle(UEP) proposed in can be used to construct such tight frames. The unitary principle says that as long as the

filters satisfy the full UEP condition, the wavelet system generated by the framelets corresponding to these filters form a wavelet tight frames in function space $L_2(\mathbb{R})$. There are many variants of the unitary extension principle, one of the full UEP conditions particular relevant to the setting in this paper is the following:

$$\sum_{i=1}^{m} \sum_{n \in \mathbb{Z}^d} a_i(k+n) a_i(n) = \delta_k, \forall k \in \mathbb{Z} \tag{1}$$

is equivalent to $W^T W_I$. For example, the linear B-spline wavelet tight frame used in many image restoration tasks is constructed via the UEP. Its associated tree filters are :

$$a_1 = \frac{1}{4}(1,2,1)^T; \quad a_2 = \frac{\sqrt{2}}{4}(1,0,-1)^T; \quad a_3 = \frac{1}{4}(-1,2,-1)^T.$$

Once the 1D framelet filter $\{a_i\}_{i=1}^m$ for generating a tight frame for $l_2(\mathbb{Z})$ is constructed, one way to generate higher dimensional tight frames is to use tensor products of 1D filters.

3.2 Adaptive Construction

For a particular class of images, there are many wavelet tight frames for us to choose from. Although they all provide perfect reconstruction of the images, some of them may yield sparser representations over the rest. Therefore, we propose the following model to find the wavelet tight frame that provides the sparest representation:

$$\min_{a_1,\cdots,a_m} \sum_j \|a_j(-\cdot) * x\|_0 \quad \text{subject to } \{a_i\}_{i=1}^m \text{ satiefies UEP condition (1)} \tag{2}$$

For signals with noise, it is natural to consider the modification:

$$\min_{a_1,\cdots,a_m,v^1,\cdots,v^m} \sum_{j=1}^{m} \|v^j - a_j(-\cdot) * x\|_2^2 + \lambda \sum_{j=1}^{m} \|v^j\|_1 \quad \text{s.t. } \{a_i\}_{i=1}^m \text{ satiefies UEP condition (1)} \tag{3}$$

Another modification is also reasonable, in fact, may yield better results sometimes.

$$\min_{a_1,\cdots,a_m} \sum_j \|a_j(-\cdot) * x\|_1 \quad \text{s.t. } \{a_i\}_{i=1}^m \text{ satiefies UEP condition (1)} \tag{4}$$

In the following, we mainly focus on (4) and its modifications. A special case of (3) is also considered. Before we proceed to the algorithms, we make a few comments about this model.

As both models involve optimization over both a and v and is non-convex, it is typically solved using alternating directions. As a result, generally, global minimum may not be obtained. Surprisingly, the local minimum obtained by alternating directions is often good enough in sparse coding and dictionary learning models.

3.3 Numerical Methods

Both model (3) and (4) involve optimization with respect to the full UEP condition, which is a quadratic constraint on the filter coefficients, we use interior point method to solve the problem. As model (3) involves optimizing over both v and a, we use an alternating direction method. This involves iterating the following two steps:

1. Given the filter coefficients, we solve (3) with respect to v only to get the sparse coefficients. That is
$$v^k := \arg\min_u \|u - a_k(-\cdot) * x\|_2^2 + \lambda \|u\|_p$$
2. Given the wavelet frame coefficients v, we update the filters. That is
$$\{a_i\}_{i=1}^m := \arg \min_{\{a_i\}_{i=1}^m} \sum_{j=1}^{m} \|v^j - a_j(-\cdot) * x\|_2^2 \quad \text{s.t. } \{a_i\}_{i=1}^m \text{ satiefies UEP condition (1)}$$

After some iterations, the above procedure is supposed to be converging to a local minimum and we obtain the adaptive filters as soon as we know how to solve the two sub-programs.

This optimization over v has an explicit solution for $p = 0, 1$, known as the hard-thresholding and soft-thresholding operation. The hard-thresholding operator is defined as

$$H_\lambda v(n) = \begin{cases} v(n), \text{ if } |v(n)| > \lambda \\ 0 \text{ otherwise} \end{cases}$$

and the soft-thresholding operator is defined as

$$S_\lambda v(n) = \begin{cases} v(n) - \lambda, \text{ if } |v(n)| > \lambda \\ 0 \text{ otherwise} \end{cases}$$

Therefore, the solution of v is $v^j := H_{\lambda/2}(a_j(-\cdot) * x$ if $p = 0$ or $v^j := S_{\lambda/2}(a_j(-\cdot) * x)$ if $p = 1$.

The optimization over a is more complex, recognizing it as a quadratic constrained optimization, we use the interior point method to solve it. During numerical experiments, we also find that simple penalty method also works well.

3.4 Discussion on the one layer transform

Once we obtain the adaptive wavelet filter, it could just be used like the normal redundant wavelet filters. That said, where wavelet transform is used, adaptive wavelet filter could also be used. For example, image compression, denoising, segmetntation, etc.

4 Compare Dictionary Learning and Model A

As we will see in this section, the dictionary learning models and model A share some similarities and have some differences. We make the following remarks:

1. Model A is convolutional form of dictionary learning. Consider the training procedure of dictionary learning, given an image or a set of images, we collect all $r \times r$ non-overlapping patches and form them into a data matrix X, then we solve the following minimization program:

$$\min_{D,C} \|X - DC\|_2^2 + \lambda \|C\|_1. \tag{5}$$

The underlying rationale is at some scale, say r, the image patch can be approximated by a sparse combination of some basis functions, where each column of D represents a base function and each column of C represents the combination coefficients for a particular patch. Consider model A in the one layer case, the objective function is

$$\min_{a_1,\cdots,a_m,v^0\cdots,v^m} \|x - \sum_{i=1}^{m} a_i * v^i\|_2^2 + \lambda \sum_{i=1}^{m} \|v^i\|_1 \tag{6}$$

With some reorganization, this can be brought into a form that is very similar to dictionary learning. Let X be the matrix formed by collecting all overlapping $r \times r$ images from an image, where each column of X represents a vectorized image patch. Let $A = (vec(a_1), \cdots, vec(a_m))$, and V is coefficient matrix which is reshaped conformally. Then the two terms in equation (6) are equivalent to :

$$\min_{A,V} \|X - AV\|_2^2 + \lambda \|V\|_1. \tag{7}$$

This is exactly the same form as dictionary learning except for that in dictionary learning, the X is formed from non-overlapping patches while in model A, X is formed from overlapping patches. If we accept the premise that image patches are translation invariant at a certain small scale, then convolutional models are a natural choice. In fact, for the purpose of finding local basis, we could form X by sampling randomly uniformly from the image. It should not be hard to show when the sample size is large, the resulted basis converges to the one obtained by solving (6). The way dictionary learning forms the matrix X corresponds to uniform sampling from a special non-lapping grid, hence a proper subset of all $r \times r$ image patches, which is unnatural. There are subsequent works on dictionary learning forming matrix X by collecting two or more sets of non-overlapping image patches, which is more close to the translation invariance assumption.

Despite the formal similarities, there are usually striking difference between the two models in practice. For example, In the practice of training dictionary learning models, people often use very large number of dictionary atoms(256 or more), where as in convolutional bases, we only use a small number of basis function(usually no more than 32). The patch based reconstruction results visually unpleasant block effect, which needs to be removed by post processing, while model A does not have such an issue.(But critically sampled model may or may not suffer from block effect depending on the support of the filter.)

2. Model A generalizes to multiple layers. Being able to generalize to multiple layers of transform is crucial to a multi-scale representation. However, dictionary learning, in its original form, cannot achieve this objective. In dictionary learning, a dictionary is obtained and each image patch is associated with a few combination coefficients. As these coefficients are unordered and are of different length for different image patches, there is no obvious way how to continue learning the pattern of the coefficients in a similar "dictionary learning" fashion. In comparison, in model A, the coefficients associated with the image are still placed on a regular grid, which, after downsampling, are still of the same format as the input, which facilitates further learning in almost the same way. It is exactly this feature that allows model A to operates on multiple scales.

3. Computationally, the analysis based variant is favorable to dictionary learning. When the parameters of both models are trained and to be used for inference, the computational costs are different. To infer the codes of a new incoming signal, dictionary learning performs a sparse coding, common procedures include matching pursuit, orthogonal matching pursuit and some path following algorithms, and they are of different computational complexity. To solve model A in the synthesis formulation would require the same computation. However, there is an analysis based variant of model, which is far more efficient computationally in that it requires only a convolution plus a point-wise operation(such as thresholding). Compared with algorithms for solving sparse coding, there is a dramatic performance gain.

5 An Analysis Based Approach to Multiscale Representation

In this section, we consider the possibility of constructing multi-scale representations of signals using analysis based approach. Such an approach has the advantage of fast forward computation. The main computation involved is merely convolution and point wise operations, no iterative procedures are needed. We don't know how to construct such a model yet. The key difficulty is we have to consider three aspects simultaneously: first, the reconstruction error should be small; two, the forward computation should be fast; three, the down sampling procedure must be explicitly incorporated to allow further operations at larger scales. A candidate is the following:

$$\min_{a,v} \sum_j \|v^j - \sum_i a_{ij} * x^i\|_2^2 + \eta \sum_i \|x^i - \sum_j b_{ij} * u^j\|_2^2 + \lambda \sum_j \|v^j\|_1 \quad \text{s.t.} \quad u^j =\uparrow\downarrow v^j \tag{8}$$

The third term needs more explanation, which we postpone to later sections.

The algorithm we use to solve this optimization problem is conjugate gradient and FISTA.

6 Multi-scale Representations

6.1 An interesting phenomenon

In Mallat's construction of wavelets, associated with every set of wavelets, there is a scaling function, which corresponds to a low frequency filter. Such a construction from multi-resolution analysis has the major benefit of fast transforms. In the one layer transform, signals are convoluted with the low frequency filter and the high frequency filters. Where as coefficients corresponding to the low frequency filter provide a crude approximation of the original signal, the coefficients corresponds to the wavelets provides information about the missing details. If we want to perform multiple layers of transformation, we simply convolve the low frequency coefficients with the same set of filters again. Continuing this way, we obtain a description of the signals at different scales.

The existence of a single low frequency filter is crucial in this procedure, since it enables fast transforms. However, in the adaptive wavelet construction, there is no explicit requirement that there must be one low frequency filter. Indeed, if we want to use the learned wavelet the same way we use pre-defined redundant wavelet tight frames, this seems necessary. What we observe in numerical experiments is very interesting. For many datasets, there is exactly one filter that sums up to 1, corresponds to the low frequency filter and each of the other filters sum up to 0, corresponds to the wavelets. This is achieved even though we didn't explicitly incorporate such a requirement in the optimization procedure. This phenomenon is persistent regardless of the initializations. However, such phenomenon is not present in all datasets. The datasets that we observed this phenomenon include: Yale human face datsets, caltech-101, finger print dataset,and many natural photos. The datasets failed to present this phenomenon include the mnist.

This phenomenon also gives an intuitive evidence why wavelet is useful in some sense. Indeed, the learned filters provide a sparse representation of signals, which automatically becomes one low frequency filter and a set of high frequency filters, wavelet has the same structure, hence may perform well, even though at that time, we don't yet know the structure of filters that provides the sparsest representation coincides with the wavelets. It is natural to ask, for what kind of data structures can we observer such a phenomenon?

6.2 Three Constructions

In previous sections, we introduced the construction of adaptive wavelet filters,which is suitable for one layer transform. Now, we want to extend the construction to multiple-layer transforms. We consider three possible structures.

The first one corresponds to the paradigm of pre-defined wavelets as illustrated in Figure 1. The root node represents the input image, each child node represents the coefficients after the convolution of the image with different filters. Higher level coefficients are obtained by convolve the low frequency coefficients in the previous layer with the same filters. As we ascending the layers, we get increasingly crude approximations of the input image. The major advantage of this paradigm is computational efficiency. Indeed, for an input signal of length N, it requires at most $O(N \log(N))$ operations to complete the multi-layer transform.

This paradigm requires the existence of a unique low frequency filter. If we are lucky, due to the phenomenon described in previous section, we may get exactly one low frequency filter for the dataset we use. If so, we can then use the learned filters exactly the same way we use pre-defined wavelet filters. We get better representation without losing any computational efficiency.

For some datasets, we may not be so lucky, there might be multiple low frequency filters, in that case, this paradigm is not applicable.

The second construction is more direct. The one layer transform is the same as the previous construction, when we go for two level transform, we convolve all of the filter coefficients, instead of the only the low frequency coefficients, with the same filter banks, the resulted structure is a full $m-$tree. Because the nodes grows exponentially as we ascend the layers, in practical applications, some care are needed to get a satisfactory performance. We will describe some of the things we need to pay attention to in the example section.

There is yet another construction as illustrated in Figure 3. In this construction, the coefficients of the next layer is obtained by first convolve the coefficients in the previous layer with different filters and then sum them up. As a result, the number of nodes can be kept constant as we ascend the layers.

7 Analysis Based Approach Linear Model

7.1 Building blocks

In this section, we consider how to extend the previous construction of adaptive wavelet tight frames to multiple layers using the third architecture.

The architecture consists of one input layer and a few intermediate layers. The filters for the first layers can be learned as described before, because the following layers are of the same structure, hence we focus on one particular layer.

The intermediate layer accepts a few sets of coefficients as input, and produces another sets of coefficients of smaller size as output. The input and output can be thought of as multi-channel images. The goal is still to produce a sparse approximation of the input coefficients. One can derive the full UEP condition for this case, but since explicitly incorporating the UEP condition involves a relative large number of constraints, hence is not quite feasible if solved using a normal optimization procedure.(It could be done though, just takes some time), we instead consider the following optimization program directly:

$$\begin{aligned} \min_{a_1,\cdots,a_m,v_1,\cdots,v_m} & \sum_i \|x_i - \sum_j a_{ij} * u_j\|_2 + \lambda \sum_j \|v_j\|_1 \\ \text{s.t.} \quad & v_j = \sum_i a_{ij}(-\cdot) * x_i \\ & u_j = \uparrow\downarrow v_j, \forall j \end{aligned} \tag{9}$$

where $\downarrow$ means down sampling and $\uparrow$ means up sampling.

The numerical algorithm for solving this optimization program is detailed in the next subsection. Some comments are in order.

First,we have replaced the full UEP condition with the reconstruction error. One has reason to doubt the minimizer of (9) may not satisfy the UEP condition at all. A necessary and sufficient condition for the filters to satisfy the UEP condition is that the reconstruction error term be error for all signals in $l_2(\mathbb{Z}^2)$. But here, even in the case when $\lambda = 0$, it is not clear that the reconstruction error term should be 0 at all, and the reconstruction error for a particular signal class is 0 may not be sufficient to guarantee the same would hold for all signal classes.

Indeed, we arrive at this model by experiments. Originally, we used interior point method to solve the following model for a sequence of decreasing η_n.

$$\begin{aligned} \min_{a_1,\cdots,a_m} & \sum_j \|v_j\|_1 \\ \text{s.t.} \quad & v_j = \sum_i a_{ij}(-\cdot) * x_i \\ & \|y_i - \sum_j a_{ij} * a_{ij}(-\cdot) y_i\|_2 \le \eta_n, \quad \forall i \end{aligned} \tag{10}$$

where the y_i are random gaussian signals.

With η_n decreasing to 0 for sufficiently large number of random gaussian signals, the full UEP condition is reinforced.

This should be the natural way to do it. Interestingly, what we found in our experiments is that

- We don't need to use gaussian random variables, the original signal works just as fine if the signal has sufficient length.
- We don't need to solve a sequence of optimization programs, as long as λ is not too large, the reconstruction error is always very close to 0, and as a result, the learned filters approximately satisfy the full UEP condition with high precision. In fact, we have the following lemma.

Lemma 1. *Let the minimizer of* (3) *be* a^**, then for any* a *satisfying* $\sum_i \|x_i - \sum_j a_{ij} * u_j\|_2 \le \sum_i \|x_i - \sum_j a^*_{ij} * u_j\|_2$*,*

$$\sum_j \|v^j\|_1 \ge \sum_j \|(v^*)^j\|_1$$

with $v^j = \sum_i a_{ij}(-\cdot) * x_i$*, and* $(v^*)^j = \sum_i a^*_{ij}(-\cdot) * x_i$*.*

Lemma 1 states that the minimizer a^*_{ij} of (3) gives the sparest representation of the input signal among all filters whose reconstruction error is no greater than the reconstruction error of a^*_{ij}. In image processing applications, we usually have a pre-defined reconstruction error tolerance in mind,

then we find the filters that gives the sparest approximation among all filters whose reconstruction error in within the predefined tolerance bound.

The second issue is the introducing of down sampling and up sampling in the model explicitly. This could be omitted if we are only interested in linear down-sampling and up-sampling procedures. But explicitly incorporating this procedure grants us more flexibility other than linear sampling procedures, for example, max pooling, which we will explore later in the construction of non-linear multi-scale representations.

7.2 Stacking the units

Once we have the building blocks, we could stack them together to get a multi-layer structure. Each layer gives a representation that is at higher scale than the last, yet tries to approximates the input signal. As the filters at each layer approximately satisfy the full UEP condition, the stacked multiple layers also jointly approximately satisfy the full UEP condition.

The top-down structure of the model makes it easy to visualize what is learned at each layer. Simply projecting back coefficients at a particular layer back to the input image space, we can see what it represents.

The nonlinearity of the model comes from two aspects. The first is the sparsity inducing thresholding operations, the second is the non-linear sampling procedures. If linear downsampling and up-sampling is used, the only source of nonlinearity is the thresholding operations.

8 Numerical Illustrations

In this section, we provide examples for the possible applications of the representation developed in the previous sections. Examples in image compression, denoising, classification will be given.

9 Discussions

In this paper, we introduced a multi-scale adaptive representation of signals based on adaptive wavelet analysis. Demonstrations of possible applications are given to show the potential use of this tool. Obviously, there are some interesting questions remain unanswered: the phase transition as we increase the value of λ, for example. Due to the length constraint of this paper, we omitted several interesting aspects of the problem, for example, the robustness properties of the model to small deformations and translations. Mallat's scattering transform is constructed using the second architecture, I wonder if the third construction also has the same robustness property, such as robust to translations and deformations?