Sparse Image Representation with Epitomes

Louise Benoît[1,3] Julien Mairal[2,3] Francis Bach[2,4] Jean Ponce[1,3]

[1]École Normale Supérieure
45, rue d'Ulm,
75005 Paris, France.

[2]INRIA
23, avenue d'Italie,
75013 Paris, France.

Abstract

Sparse coding, which is the decomposition of a vector using only a few basis elements, is widely used in machine learning and image processing. The basis set, also called dictionary, is learned to adapt to specific data. This approach has proven to be very effective in many image processing tasks. Traditionally, the dictionary is an unstructured "flat" set of atoms. In this paper, we study structured dictionaries [1] which are obtained from an epitome [11], or a set of epitomes. The epitome is itself a small image, and the atoms are all the patches of a chosen size inside this image. This considerably reduces the number of parameters to learn and provides sparse image decompositions with shift-invariance properties. We propose a new formulation and an algorithm for learning the structured dictionaries associated with epitomes, and illustrate their use in image denoising tasks.

1. Introduction

Jojic, Frey and Kannan [11] introduced in 2003 a probabilistic generative image model called an *epitome*. Intuitively, the epitome is a small image that summarizes the content of a larger one, in the sense that for any patch from the large image there should be a similar one in the epitome. This is an intriguing notion, which has been applied to image reconstruction tasks [11], and epitomes have also been extended to the video domain [5], where they have been used in denoising, superresolution, object removal and video interpolation. Other successful applications of epitomes include location recognition [20] or face recognition [6].

Aharon and Elad [1] have introduced an alternative formulation within the sparse coding framework called *image-signature dictionary*, and applied it to image denoising. Their formulation unifies the concept of epitome and dictionary learning [9, 21] by allowing an image patch to be represented as a sparse linear combination of several patches extracted from the epitome (Figure 1). The resulting sparse representations are highly redundant (there are as many dictionary elements as overlapping patches in the epitome), with dictionaries represented by a reasonably small number of parameters (the number of pixels in the epitome). Such a representation has also proven to be useful for texture synthesis [22].

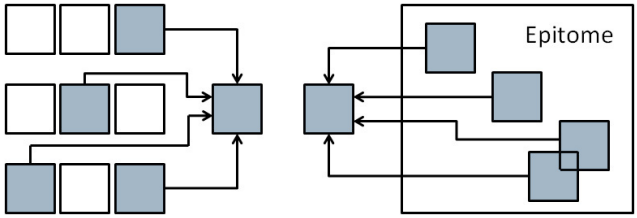

Figure 1. A "flat" dictionary (left) vs. an epitome (right). Sparse coding with an epitome is similar to sparse coding with a flat dictionary, except that the atoms are extracted from the epitome and may overlap instead of being chosen from an unstructured set of patches and assumed to be independent one from each other.

In a different line of work, some research has been focusing on learning shift-invariant dictionaries [13, 23], in the sense that it is possible to use dictionary elements with different shifts to represent signals, exhibiting patterns that may appear several times at different positions. While this is different from the image-signature dictionaries of Aharon and Elad [1], the two ideas are related, and as shown in this paper, such a shift invariance can be achieved by using a collection of smaller epitomes. In fact, one of our main contributions is to unify the frameworks of epitome and dictionary learning, and establish the continuity between dictionaries, dictionaries with shift invariance, and epitomes.

We propose a formulation based on the concept of epitomes/image-signature-dictionaries introduced by [1, 11], which allows to learn a collection of epitomes, and which is generic enough to be used with epitomes that may

[3]WILLOW project-team, Laboratoire d'Informatique de l'École Normale Supérieure, ENS/INRIA/CNRS UMR 8548.

[4]SIERRA team, Laboratoire d'Informatique de l'École Normale Supérieure, ENS/INRIA/CNRS UMR 8548.

have different shapes, or with different dictionary parameterizations. We present this formulation for the specific case of image patches for simplicity, but it applies to spatio-temporal blocks in a straightforward manner.

The following notation is used throughout the paper: we define for $q \geqslant 1$ the ℓ_q-norm of a vector $\mathbf{x}$ in $\mathbb{R}^m$ as $\|\mathbf{x}\|_q \triangleq (\sum_{j=1}^m |x_j|^q)^{1/q}$, where x_j denotes the j-th coordinate of $\mathbf{x}$. if $\mathbf{X}$ is a matrix in $\mathbb{R}^{m\times n}$, $\mathbf{x}^i$ will denote its ith row, while $\mathbf{x}_j$ will denote its jth column. As usual, $x_{i,j}$ will denote the entry of $\mathbf{X}$ at the ith-row and j^{th}-column. We consider the Frobenius norm of $\mathbf{X}$: $\|\mathbf{X}\|_F \triangleq (\sum_{i=1}^m \sum_{j=1}^n x_{i,j}^2)^{1/2}$.

This paper is organized as follows: Section 2 introduces our formulation. We present our dictionary learning algorithm in Section 3. Section 4 introduces different improvements for this algorithm, and Section 5 demonstrates experimentally the usefulness of our approach.

2. Proposed Approach

Given a set of n training image patches of size m pixels, represented by the columns of a matrix $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$ in $\mathbb{R}^{m\times n}$, the classical dictionary learning formulation, as introduced by [21] and revisited by [9, 14], tries to find a dictionary $\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_p]$ in $\mathbb{R}^{m\times p}$ such that each signal $\mathbf{x}_i$ can be represented by a sparse linear combination of the columns of $\mathbf{D}$. More precisely, the dictionary $\mathbf{D}$ is learned along with a matrix of decomposition coefficients $\mathbf{A} = [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_n]$ in $\mathbb{R}^{p\times n}$ such that $\mathbf{x}_i \approx \mathbf{D}\boldsymbol{\alpha}_i$ for every signal $\mathbf{x}_i$. Following [14], we consider the following formulation:

$$\min_{\mathbf{D}\in\mathcal{D},\mathbf{A}\in\mathbb{R}^{p\times n}} \frac{1}{n}\sum_{i=1}^n \Big[\frac{1}{2}\|\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i\|_2^2 + \lambda\|\boldsymbol{\alpha}_i\|_1\Big], \quad (1)$$

where the quadratic term ensures that the vectors $\mathbf{x}_i$ are close to the approximation $\mathbf{D}\boldsymbol{\alpha}_i$, the ℓ_1-norm induces sparsity in the coefficients $\boldsymbol{\alpha}_i$ (see, e.g., [4, 24]), and λ controls the amount of regularization. To prevent the columns of $\mathbf{D}$ from being arbitrarily large (which would lead to arbitrarily small values of the $\boldsymbol{\alpha}_i$), the dictionary $\mathbf{D}$ is constrained to belong to the convex set $\mathcal{D}$ of matrices in $\mathbb{R}^{m\times p}$ whose columns have an ℓ_2-norm less than or equal to one:

$$\mathcal{D} \triangleq \{\mathbf{D} \in \mathbb{R}^{m\times p} \text{ s.t. } \forall j = 1, \ldots, p,\ \|\mathbf{d}_j\|_2 \leqslant 1\}.$$

As will become clear shortly, this constraint is not adapted to dictionaries extracted from epitomes, since overlapping patches cannot be expected to all have the same norm. Thus we introduce an unconstrained formulation equivalent to Eq. (1):

$$\min_{\substack{\mathbf{D}\in\mathbb{R}^{m\times n},\\ \mathbf{A}\in\mathbb{R}^{p\times n}}} \frac{1}{n}\sum_{i=1}^n \Big[\frac{1}{2}\|\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i\|_2^2 + \lambda\sum_{j=1}^p \|\mathbf{d}_j\|_2|\alpha_{j,i}|\Big]. \quad (2)$$

This formulation removes the constraint $\mathbf{D} \in \mathcal{D}$ from Eq. (1), and replaces the ℓ_1-norm by a weighted ℓ_1-norm. As shown in Appendix A, Eq. (1) and Eq. (2) are equivalent in the sense that a solution of Eq. (1) is also solution of Eq. (2), and for every solution of Eq. (2), a solution for Eq. (1) can be obtained by normalizing its columns to one. To the best of our knowledge, this equivalent formulation is new, and is key to learning an epitome with ℓ_1-regularization: the use of a convex regularizer (the ℓ_1-norm) that empirically provides better-behaved dictionaries than ℓ_0 (where the ℓ_0 pseudo-norm counts the number of non-zero elements in a vector) for denoising tasks (see Table 1) differentiates us from the ISD formulation of [1]. To prevent degenerate solutions in the dictionary learning formulation with ℓ_1-norm, it is important to constrain the dictionary elements with the ℓ_2-norm. Whereas such a constraint can easily be imposed in classical dictionary learning, its extension to epitome learning is not straightforward, and the original ISD formulation is not compatible with convex regularizers. Eq. (2) is an equivalent unconstrained formulation, which lends itself well to epitome learning.

We can now formally introduce the general concept of an epitome as a small image of size $\sqrt{M} \times \sqrt{M}$, encoded (for example in row order) as a vector $\mathbf{E}$ in $\mathbb{R}^M$. We also introduce a linear operator $\varphi : \mathbb{R}^M \to \mathbb{R}^{m\times p}$ that extracts all overlapping patches from the epitome $\mathbf{E}$, and rearranges them into the columns of a matrix of $\mathbb{R}^{m\times p}$, the integer p being the number of such overlapping patches. Concretely, we have $p = (\sqrt{M} - \sqrt{m} + 1)^2$. In this context, $\varphi(\mathbf{E})$ can be interpreted as a traditional flat dictionary with p elements, except that it is generated by a small number M of parameters compared to the pm parameters of the flat dictionary. Our approach thus generalizes to a much wider range of epitomic structures using any mapping φ that admits fast projections on $\mathrm{Im}(\varphi)$. The functions φ we have used so far are relatively simple, but give a framework that easily extends to families of epitomes, shift-invariant dictionaries, and plain dictionaries. The only assumption we make is that φ is a linear operator of rank M (i.e., φ is injective). This list is not exhaustive, which naturally opens up new perspectives. The fact that a dictionary $\mathbf{D}$ is obtained from an epitome is characterized by the fact that $\mathbf{D}$ is in the image $\mathrm{Im}\,\varphi$ of the linear operator φ. Given a dictionary $\mathbf{D}$ in $\mathrm{Im}\,\varphi$, the unique (by injectivity of φ) epitome representation can be obtained by computing the inverse of φ on $\mathrm{Im}\,\varphi$, for which a closed form using pseudo-inverses exists as shown in Appendix B.

Our goal being to adapt the epitome to the training image patches, the general minimization problem can therefore be expressed as follows:

$$\min_{\substack{\mathbf{D}\in\mathrm{Im}\,\varphi,\\ \mathbf{A}\in\mathbb{R}^{p\times n}}} \frac{1}{n}\sum_{i=1}^n \Big[\frac{1}{2}\|\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i\|_2^2 + \lambda\sum_{j=1}^p \|\mathbf{d}_j\|_2|\alpha_{j,i}|\Big]. \quad (3)$$

There are several motivations for such an approach. As discussed above, the choice of the function φ lets us adapt this technique to different problems such as multiple epitomes or any other type of dictionary representation. This formulation is therefore deliberately generic. In practice, we have mainly focused on two simple cases in the experiments of this paper: a single epitome [11] (or image signature dictionary [1]) and a set of epitomes. Furthermore, we have now come down to a more traditional, and well studied problem: dictionary learning. We will therefore use the techniques and algorithms developed in the dictionary learning literature to solve the epitome learning problem.

3. Basic Algorithm

As for classical dictionary learning, the optimization problem of Eq. (3) is not jointly convex in $(\mathbf{D}, \mathbf{A})$, but is convex with respect to $\mathbf{D}$ when $\mathbf{A}$ is fixed and vice-versa. A block-coordinate descent scheme that alternates between the optimization of $\mathbf{D}$ and $\mathbf{A}$, while keeping the other parameter fixed, has emerged as a natural and simple way for learning dictionaries [9, 10], which has proven to be relatively efficient when the training set is not too large. Even though the formulation remains nonconvex and therefore this method is not guaranteed to find the global optimum, it has proven experimentally to be good enough for many tasks [9].

We therefore adopt this optimization scheme as well, and detail the different steps below. Note that other algorithms such as stochastic gradient descent (see [1, 14]) could be used as well, and in fact can easily be derived from the material of this section. However, we have chosen not to investigate these kind of techniques for simplicity reasons. Indeed, stochastic gradient descent algorithms are potentially more efficient than the block-coordinate scheme mentioned above, but require the (sometimes non-trivial) tuning of a learning rate.

3.1. Step 1: Optimization of $\mathbf{A}$ with $\mathbf{D}$ Fixed.

In this step of the algorithm, $\mathbf{D}$ is fixed, so the constraint $\mathbf{D} \in \operatorname{Im}\varphi$ is not involved in the optimization of $\mathbf{A}$. Furthermore, note that updating the matrix $\mathbf{A}$ consists of solving n independent optimization problems with respect to each column $\boldsymbol{\alpha}_i$. For each of them, one has to solve a weighted-ℓ_1 optimization problem. Let us consider the update of a column $\boldsymbol{\alpha}_i$ of $\mathbf{A}$.

We introduce the matrix $\boldsymbol{\Gamma} \triangleq \operatorname{diag}[\|\mathbf{d}_1\|_2, .., \|\mathbf{d}_p\|_2]$, and define $\mathbf{D}' = \mathbf{D}\boldsymbol{\Gamma}^{-1}$. If $\boldsymbol{\Gamma}$ is non-singular, we show in Appendix A that the relation $\boldsymbol{\alpha}_i'^{\star} = \boldsymbol{\Gamma}\boldsymbol{\alpha}_i^{\star}$ holds, where

$$\boldsymbol{\alpha}_i'^{\star} = \operatorname*{argmin}_{\boldsymbol{\alpha}_i' \in \mathbb{R}^p} \frac{1}{2}\|\mathbf{x}_i - \mathbf{D}'\boldsymbol{\alpha}_i'\|_F^2 + \lambda\|\boldsymbol{\alpha}_i'\|_1, \quad \text{and}$$

$$\boldsymbol{\alpha}_i^{\star} = \operatorname*{argmin}_{\boldsymbol{\alpha}_i \in \mathbb{R}^p} \frac{1}{2}\|\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i'\|_F^2 + \lambda\sum_{j=1}^{p}\|\mathbf{d}_j\|_2|\alpha_{j,i}|.$$

This shows that the update of each column can easily be obtained with classical solvers for ℓ_1-decomposition problems. We use to that effect the LARS algorithm [8], implemented in the software accompanying [14].

Since our optimization problem is invariant by multiplying $\mathbf{D}$ by a scalar and $\mathbf{A}$ by its inverse, we then proceed to the following renormalization to ensure numerical stability and prevent the entries of $\mathbf{D}$ and $\mathbf{A}$ from becoming too large: we rescale $\mathbf{D}$ and $\mathbf{A}$ with

$$s = \min_{j \in [\![1,n]\!]} \|\mathbf{d}_j\|_2, \text{ and define } \mathbf{D} \leftarrow \frac{1}{s}\mathbf{D} \text{ and } \mathbf{A} \leftarrow s\mathbf{A}.$$

Since the image of φ is a vector space, $\mathbf{D}$ stays in the image of φ after the normalization. And as noted before, it does not change the value of the objective function.

3.2. Step 2: Optimization of $\mathbf{D}$ with $\mathbf{A}$ Fixed.

We use a projected gradient descent algorithm [3] to update $\mathbf{D}$. The objective function f minimized during this step can be written as:

$$f(\mathbf{D}) \triangleq \frac{1}{2}\|\mathbf{X} - \mathbf{D}\mathbf{A}\|_F^2 + \lambda\sum_{j=1}^{p}\|\mathbf{d}_j\|_2\|\boldsymbol{\alpha}^j\|_1, \quad (4)$$

where $\mathbf{A}$ is fixed, and we recall that $\boldsymbol{\alpha}^j$ denotes its j-th row. The function f is differentiable, except when a column of $\mathbf{D}$ is equal to zero, which we assume without loss of generality not to be the case. Suppose indeed that a column $\mathbf{d}_j$ of $\mathbf{D}$ is equal to zero. Then, without changing the value of the cost function of Eq. (3), one can set the corresponding row $\boldsymbol{\alpha}^j$ to zero as well, and it results in a function f defined in Eq. (4) that does not depend on $\mathbf{d}_j$ anymore. We have, however, not observed such a situation in our experiments.

The function f can therefore be considered as differentiable, and one can easily compute its gradient as:

$$\nabla f(\mathbf{D}) = -(\mathbf{X} - \mathbf{D}\mathbf{A})\mathbf{A}^T + \mathbf{D}\boldsymbol{\Delta},$$

where $\boldsymbol{\Delta}$ is defined as $\boldsymbol{\Delta} \triangleq \operatorname{diag}(\lambda\frac{\|\boldsymbol{\alpha}^1\|_1}{\|\mathbf{d}_1\|_2}, \ldots, \lambda\frac{\|\boldsymbol{\alpha}^p\|_1}{\|\mathbf{d}_p\|_2})$.

To use a projected gradient descent, we now need a method for projecting $\mathbf{D}$ onto the convex set $\operatorname{Im}\varphi$, and the update rule becomes:

$$\mathbf{D} \leftarrow \Pi_{\operatorname{Im}\varphi}[\mathbf{D} - \rho\nabla f(\mathbf{D})],$$

where $\Pi_{\operatorname{Im}\varphi}$ is the orthogonal projector onto $\operatorname{Im}\varphi$, and ρ is a gradient step, chosen with a line-search rule, such as the Armijo rule [3].

Interestingly, in the case of the single epitome (and in fact in any other extension where φ is a linear operator that extracts some patches from a parameter vector $\mathbf{E}$), this projector admits a closed form: let us consider the linear operator $\varphi^* : \mathbb{R}^{m\times p} \rightarrow \mathbb{R}^M$, such that for a matrix $\mathbf{D}$ in $\mathbb{R}^{m\times p}$,

a pixel of the epitome $\varphi^*(\mathbf{D})$ is the average of the entries of $\mathbf{D}$ corresponding to this pixel value. We give the formal form of this operator in Appendix B, and show the following results:

(i) φ^* is indeed linear,

(ii) $\Pi_{\mathrm{Im}\,\varphi} = \varphi \circ \varphi^*$.

With this closed form of $\Pi_{\mathrm{Im}\,\varphi}$ in hand, we now have an efficient algorithmic procedure for performing the projection. Our method is therefore quite generic, and can adapt to a wide variety of functions φ. Extending it when φ is not linear, but still injective and with an efficient method to project on $\mathrm{Im}\,\varphi$ will be the topic of future work.

4. Improvements

We present in this section several improvements to our basic framework, which either improve the convergence speed of the algorithm, or generalize the formulation.

4.1. Accelerated Gradient Method for Updating $\mathbf{D}$.

A first improvement is to accelerate the convergence of the update of $\mathbf{D}$ using an accelerated gradient technique [2, 19]. These methods, which build upon early works by Nesterov [18], have attracted a lot of attention recently in machine learning and signal processing, especially because of their fast convergence rate (which is proven to be optimal among first-order methods), and their ability to deal with large, possibly nonsmooth problems.

Whereas the value of the objective function with classical gradient descent algorithms for solving smooth convex problems is guaranteed to decrease with a convergence rate of $O(1/k)$, where k is the number of iterations, other algorithmic schemes have been proposed with a convergence rate of $O(1/k^2)$ with the same cost per iteration as classical gradient algorithms [2, 18, 19]. The difference between these methods and gradient descent algorithms is that two sequences of parameters are maintained during this iterative procedure, and that each update uses information from past iterations. This leads to theoretically better convergence rates, which are often also better in practice.

We have chosen here for its simplicity the algorithm FISTA of Beck and Teboulle [2], which includes a practical line-search scheme for automatically tuning the gradient step. Interestingly, we have indeed observed that the algorithm FISTA was significantly faster to converge than the projected gradient descent algorithm.

4.2. Multi-Scale Version

To improve the results without increasing the computing time, we have also implemented a multi-scale approach that exploits the spatial nature of the epitome. Instead of directly learning an epitome of size M, we first learn an epitome of a smaller size on a reduced image with corresponding smaller patches, and after upscaling, we use the resulting epitome as the initialization for the next scale. We iterate this process in practice two to three times. The procedure is illustrated in Figure 2. Intuitively, learning smaller epitomes is an easier task than directly learning a large one, and such a procedure provides a good initialization for learning a large epitome.

Multi-scale Epitome Learning.

Input: n number of scales, r ratio between each scale, $\mathbf{E}_0$ random initialization for the first scale.
for $k = 1$ **to** n **do**
  Given I_k rescaling of image I for ratio $\frac{1}{r^{n-k}}$,
  $\mathbf{X}_k$ the corresponding patches,
  initialize with $\mathbf{E} = \mathrm{upscale}(\mathbf{E}_{k-1}, r)$,
  $\mathbf{E}_k$ = epitome $(\mathbf{X}_k, \mathbf{E})$.
end for
Output: learned epitome $\mathbf{E}$.

Figure 2. Multi-scale epitome learning algorithm.

4.3. Multi-Epitome Extension

Another improvement is to consider not a single epitome but a family of epitomes in order to learn dictionaries with some shift invariance, which has been the focus of recent work [13, 23]. Note that different types of structured dictionaries have also been proposed with the same motivation for learning shift-invariant features in image classification tasks [12], but in a significantly different framework (the structure in the dictionaries learned in [12] comes from a different sparsity-inducing penalization).

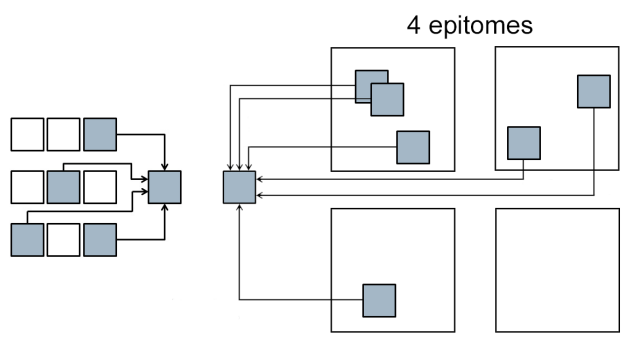

Figure 3. A "flat" dictionary (left) vs. a collection of 4 epitomes (right). The atoms are extracted from the epitomes and may overlap.

As mentioned before, we are able to learn a set of N epitomes instead of a single one by changing the function φ introduced earlier. The vector $\mathbf{E}$ now contains the pixels (parameters) of several small epitomes, and φ is the linear operator that extracts all overlapping patches from all epitomes. In the same way, the projector on $\mathrm{Im}\,\varphi$ is still easy to compute in closed form, and the rest of the algorithm stays unchanged. Other "epitomic" structures could easily be used within our framework, even though we have limited

ourselves for simplicity to the case of single and multiple epitomes of the same size and shape.

The multi-epitome version of our approach can be seen as an interpolation between classical dictionary and single epitome. Indeed, defining a multitude of epitomes of the same size as the considered patches is equivalent to working with a dictionary. Defining a large number a epitomes slightly larger than the patches is equivalent to shift-invariant dictionaries. In Section 5, we experimentally compare these different regimes for the task of image denoising.

4.4. Initialization

Because of the nonconvexity of the optimization problem, the question of the initialization is an important issue in epitome learning. We have already mentioned a multi-scale strategy to overcome this issue, but for the first scale, the problem remains. Whereas classical flat dictionaries can naturally be initialized with prespecified dictionaries such as overcomplete DCT basis (see [9]), the epitome does not admit such a natural choice. In all the experiences (unless written otherwise), we use as the initialization a single epitome (or a collection of epitomes), common to all experiments, which is learned using our algorithm, initialized with a Gaussian low-pass filtered random image, on a set of $100\,000$ random patches extracted from $5\,000$ natural images (all different from the test images used for denoising).

5. Experimental Validation

Figure 4. House, Peppers, Cameraman, Lena, Boat and Barbara images.

We provide in this section qualitative and quantitative validation. We first study the influence of the different model hyperparameters on the visual aspect of the epitome before moving to an image denoising task. We choose to represent the epitomes as images in order to visualize more easily the patches that will be extracted to form the images. Since epitomes contain negative values, they are arbitrarily rescaled between 0 and 1 for display.

In this section, we will work with several images, which are shown in Figure 4.

5.1. Influence of the Initialization

In order to measure the influence of the initialization on the resulting epitome, we have run the same experience with different initializations. Figure 5 shows the different results obtained.

The difference in contrast may be due to the scaling of the data in the displaying process. This experiment illustrates that different initializations lead to visually different epitomes. Whereas this property might not be desirable, the classical dictionary learning framework also suffers from this issue, but yet has led to successful applications in image processing [9].

Figure 5. Three epitomes obtained on the boat image for different initializations, but all the same parameters. Left: epitome obtained with initialization on a epitome learned on random patches from natural images. Middle and Right: epitomes obtained for two different random initializations.

5.2. Influence of the Size of the Patches

The size of the patches seem to play an important role in the visual aspect of the epitome. We illustrate in Figure 6 an experiment where pairs of epitome of size 46×46 are learned with different sizes of patches.

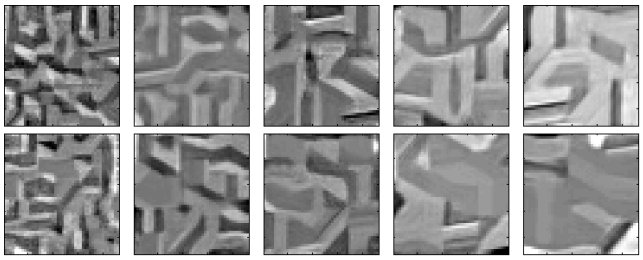

Figure 6. Pairs of epitomes of width 46 obtained for patches of width $6, 8, 9, 10$ and 12. All other parameters are unchanged. Experiments run with 2 scales (20 iterations for the first scale, 5 for the second) on the house image.

As we see, learning epitomes with small patches seems to introduce finer details and structures in the epitome, whereas large patches induce epitomes with coarser structures.

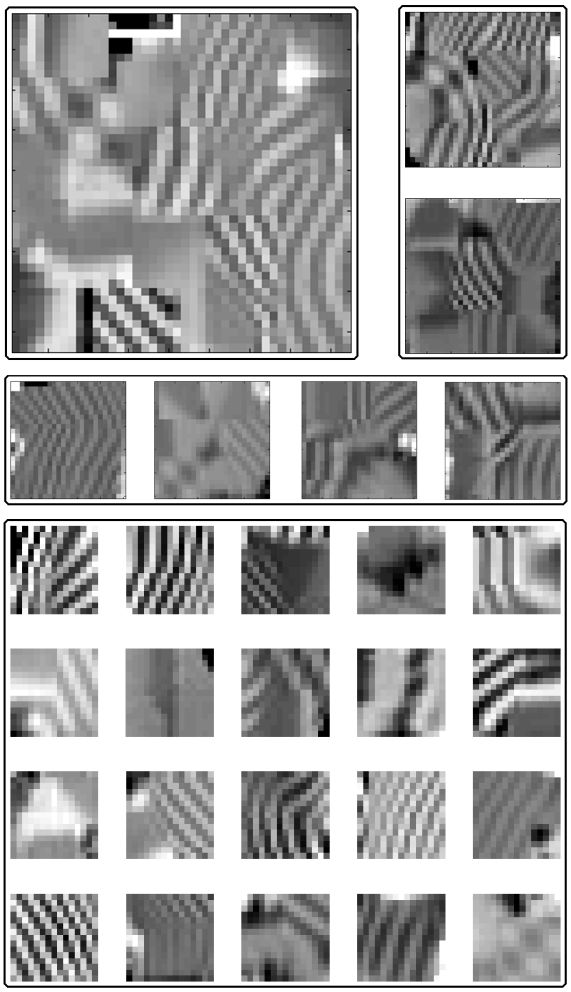

Figure 7. 1, 2, 4 and 20 epitomes learned on the barbara image for the same parameters. They are of sizes 42, 32, 25 and 15 in order to keep the same number of elements in $\mathbf{D}$. They are not represented to scale.

5.3. Influence of the Number of Epitomes

We present in this section an experiment where the number of learned epitomes vary, while keeping the same numbers of columns in $\mathbf{D}$. The 1, 2, 4 and 20 epitomes learned on the image barbara are shown in Figure 7. When the number of epitomes is small, we observe in the epitomes some discontinuities between texture areas with different visual characteristics, which is not the case when learning several independant epitomes.

5.4. Application to Denoising

In order to evaluate the performance of epitome learning in various regimes (single epitome, multiple epitomes), we use the same methodology as [1] that uses the successful denoising method first introduced by [9]. Let us consider first the classical problem of restoring a noisy image $\boldsymbol{y}$ in $\mathbb{R}^n$ which has been corrupted by a white Gaussian noise of standard deviation σ. We denote by $\boldsymbol{y}_i$ in $\mathbb{R}^m$ the patch of $\boldsymbol{y}$ centered at pixel i (with any arbitrary ordering of the image pixels).

The method of [9] proceeds as follows:

- Learn a dictionary $\mathbf{D}$ adapted to all overlapping patches $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots$ from the noisy image $\boldsymbol{y}$.
- Approximate each noisy patch using the learned dictionary with a greedy algorithm called orthogonal matching pursuit (OMP) [17] to have a clean estimate of every patch of $\boldsymbol{y}_i$ by addressing the following problem

$$\underset{\boldsymbol{\alpha}_i \in \mathbb{R}^p}{\operatorname{argmin}} \|\boldsymbol{\alpha}_i\|_0 \ \text{ s.t. } \ \|\boldsymbol{y}_i - \mathbf{D}\boldsymbol{\alpha}_i\|_2^2 \leqslant (C\sigma^2),$$

  where $\mathbf{D}\boldsymbol{\alpha}_i$ is a clean estimate of the patch $\boldsymbol{y}_i$, $\|\boldsymbol{\alpha}_i\|_0$ is the ℓ_0 pseudo-norm of $\boldsymbol{\alpha}_i$, and C is a regularization parameter. Following [9], we choose $C = 1.15$.
- Since every pixel in $\boldsymbol{y}$ admits many clean estimates (one estimate for every patch the pixel belongs to), average the estimates.

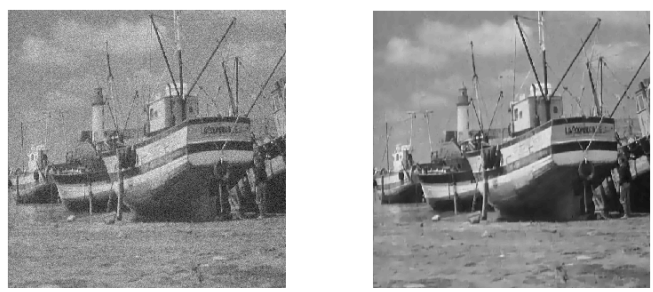

Figure 8. Artificially noised boat image (with standard deviation $\sigma = 15$), and the result of our denoising algorithm.

Quantitative results for single epitome, and multi-scale multi-epitomes are presented in Table 1 on six images and five levels of noise. We evaluate the performance of the denoising process by computing the peak signal-to-noise ratio (PSNR) for each pair of images. For each level of noise, we have selected the best regularization parameter λ overall the six images, and have then used it all the experiments. The PNSR values are averaged over 5 experiments with 5 different noise realizations. The mean standard deviation is of 0.05dB both for the single epitome and the multi-scale multi-epitomes.

We see from this experiment that the formulation we propose is competitive compared to the one of [1]. Learning multi epitomes instead of a single one seems to provide better results, which might be explained by the lack of flexibility of the single epitome representation. Evidently, these results are not as good as recent state-of-the-art denoising algorithms such as [7, 15] which exploit more sophisticated

σ		house	peppers	c.man	barbara	lena	boat
10	**IE**	35.98	**34.52**	**33.90**	**34.41**	**35.51**	**33.70**
	E	35.86	34.41	33.83	34.01	35.43	33.63
15	**IE**	**34.45**	**32.50**	**31.65**	**32.23**	**33.74**	**31.81**
	E	34.32	32.36	31.59	31.84	33.66	31.75
20	**IE**	**33.18**	**31.00**	**30.19**	**30.69**	**32.42**	**30.45**
	E	33.08	30.93	30.11	30.33	32.35	30.37
25	**IE**	**32.02**	**29.82**	**29.08**	**29.49**	**31.36**	**29.36**
	E	31.96	29.77	29.01	29.14	31.29	29.30
50	**IE**	27.83	26.06	25.57	**25.04**	**27.90**	26.01
	E	**27.83**	**26.07**	**25.60**	24.86	27.82	**26.02**

Table 1. PSNR Results. First Row: 20 epitomes of size 7×7 learned with 3 scales (**IE**: improved epitome); Second row: single epitome of size 42×42 (**E**). Best results are in bold.

σ	**IE**	**E**	[1]	[11]	[9]	[7]	[15]
10	34.83	34.67	34.71	28.83	34.76	35.24	35.32
15	32.95	32.79	32.84	28.92	32.87	33.43	33.50
20	31.55	31.41	31.36	28.55	31.52	32.15	32.18
25	30.41	30.29	29.99	28.12	30.42	31.15	31.11
50	26.57	26.52	25.91	25.21	26.66	27.69	27.87
mean	31.26	31.14	30.96	27.93	31.25	31.93	32.00

Table 2. Quantitative comparative evaluation. PSNR values are averaged over 5 images. We compare ourselves to two previous epitome learning based algorithms: ISD ([1]) and epitomes by Jojic, Frey and Kannan ([11] as reported in [1]), and to three more elaborate dictionary learning based algorithms K-SVD ([9]), BM3D ([7]), and LSSC ([15]).

image models. But our goal is to illustrate the performance of epitome learning on an image reconstruction task, in order to better understand these formulations.

6. Conclusion

We have introduced in this paper a new formulation and an efficient algorithm for learning epitomes in the context of sparse coding, extending the work of Aharon and Elad [1], and unifying it with recent work on shift-invariant dictionary learning. Our approach is generic, can interpolate between these two regimes, and can possibly be applied to other formulations. Future work will extend our framework to the video setting, to other image processing tasks such as inpainting, and to learning image features for classification or recognition tasks, where shift invariance has proven to be a key property to achieving good results [12]. Another direction we are pursuing is to find a way to encode other invariant properties through different mapping functions φ.

Acknowledgments. This work was partly supported by the European Community under the ERC grants "VideoWorld" and "Sierra".

A. Appendix: ℓ_1-Norm and Weighted ℓ_1-Norm

In this appendix, we will show the equivalence between the two minimization problems introduced in section 3.1.

Let us denote

$$F(\mathbf{D}, \boldsymbol{\alpha}) = \frac{1}{2}\|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + \frac{\lambda}{2}\sum_{j=1}^{p} \|\mathbf{d}_j\|_2 |\alpha_j|, \tag{5}$$

$$\text{and} \quad G(\mathbf{D}, \boldsymbol{\alpha}) = \frac{1}{2}\,||\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}||_2^2 + \lambda\|\boldsymbol{\alpha}\|_1. \tag{6}$$

Let us define $\boldsymbol{\alpha}' \in \mathbb{R}^p$ and $\mathbf{D}' \in \mathbb{R}^{m\times p}$ such that $\mathbf{D}' = \mathbf{D}\boldsymbol{\Gamma}^{-1}$, and $\boldsymbol{\alpha}' = \boldsymbol{\Gamma}\boldsymbol{\alpha}$, where $\boldsymbol{\Gamma} = \mathrm{diag}[\|\mathbf{d}_1\|_2, .., \|\mathbf{d}_p\|_2]$. The goal is to show that $\boldsymbol{\alpha}'^{\star} = \boldsymbol{\Gamma}\boldsymbol{\alpha}^{\star}$, where:

$$\boldsymbol{\alpha}^{\star} = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}}\, F(\mathbf{D}, \boldsymbol{\alpha}), \quad \text{and} \quad \boldsymbol{\alpha}'^{\star} = \underset{\boldsymbol{\alpha}'}{\operatorname{argmin}}\, G(\mathbf{D}', \boldsymbol{\alpha}').$$

We clearly have: $\mathbf{D}\boldsymbol{\alpha} = \mathbf{D}'\boldsymbol{\alpha}'$. Furthermore, since $\boldsymbol{\Gamma}\boldsymbol{\alpha} = \boldsymbol{\alpha}'$, we have: $\forall j = 1, \ldots, p, \quad \|\mathbf{d}_j\|_2|\alpha_j| = |\alpha'_j|$.

Therefore,

$$F(\mathbf{D}, \boldsymbol{\alpha}) = G(\mathbf{D}', \boldsymbol{\alpha}'). \tag{7}$$

Moreover, since for all $\mathbf{D}$, $\mathbf{D}'$ is in the set $\mathcal{D}$, we have shown the equivalence between Eq. (1) and Eq. (2).

B. Appendix: Projection on $\mathrm{Im}\,\varphi$

In this appendix, we will show how to compute the orthogonal projection on the vector space $\mathrm{Im}\,\varphi$. Let us denote by $\mathbf{R}_i$ the binary matrix in $\{0,1\}^{m\times M}$ that extracts the i-th patch from $\mathbf{E}$. Note that with this notation, the matrix $\mathbf{R}_i$ is a binary $M \times m$ matrix corresponding to a linear operator that takes a patch of size m and place it at the location i in an epitome of size M which is zero everywhere else. We therefore have $\varphi(\mathbf{E}) = [\mathbf{R}_1\mathbf{E}, \ldots, \mathbf{R}_p\mathbf{E}]$.

We denote by $\varphi^* : \mathbb{R}^{m\times p} \to \mathbb{R}^M$ the linear operator defined as

$$\varphi^*(\mathbf{D}) = (\sum_{j=1}^{p} \mathbf{R}_j^T\mathbf{R}_j)^{-1}(\sum_{j=1}^{p} \mathbf{R}_i^T\mathbf{D}),$$

which creates an epitome of size M such that each pixel contains the average of the corresponding entries in $\mathbf{D}$. Indeed, the $M \times M$ matrix $(\sum_{j=1}^{p} \mathbf{R}_j^T\mathbf{R}_j)^{-1}$ is diagonal and the entry i on the diagonal is the number of entries in $\mathbf{D}$ corresponding to the pixel i in the epitome.

Denoting by

$$\mathbf{R} \triangleq \begin{bmatrix} \mathbf{R}_1 \\ \vdots \\ \mathbf{R}_p \end{bmatrix},$$

which is a $mp \times M$ matrix, we have $\mathrm{vec}(\varphi(\mathbf{E})) = \mathbf{R}\mathbf{E}$, where $\mathrm{vec}(\mathbf{D}) \triangleq [\mathbf{d}_1^T, \ldots, \mathbf{d}_p^T]^T$, which is the vector of size

mp obtained by concatenating the columns of $\mathbf{D}$, and also $\varphi^*(\mathbf{D}) = (\mathbf{R}^T\mathbf{R})^{-1}\mathbf{R}^T\text{vec}(\mathbf{D})$.

Since $\text{vec}(\text{Im}\,\varphi) = \text{Im}\,\mathbf{R}$ and $\text{vec}(\varphi(\varphi^*(\mathbf{D}))) = \mathbf{R}(\mathbf{R}^T\mathbf{R})^{-1}\mathbf{R}^T\text{vec}(\mathbf{D})$, which is an orthogonal projection onto $\text{Im}\,\mathbf{R}$, it results the two following properties which are useful in our framework and classical in signal processing with overcomplete representations ([16]):

- φ^* is the inverse function of φ on $\text{Im}\,\varphi$: $\varphi^* \circ \varphi = \text{Id}$.
- $(\varphi \circ \varphi^*)$ is the orthogonal projector on $\text{Im}\,\varphi$.

References

[1] M. Aharon and M. Elad. Sparse and redundant modeling of image content using an image-signature-dictionary. *SIAM Journal on Imaging Sciences*, 1(3):228–247, July 2008.

[2] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[3] D. P. Bertsekas. *Nonlinear programming*. Athena Scientific Belmont, 1999.

[4] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1998.

[5] V. Cheung, B. Frey, and N. Jojic. Video epitomes. In *Proc. CVPR*, 2005.

[6] X. Chu, S. Yan, L. Li, K. L. Chan, and T. S. Huang. Spatialized epitome and its applications. In *Proc. CVPR*, 2010.

[7] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095, 2007.

[8] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Annals of statistics*, 32(2):407–499, 2004.

[9] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 54(12):3736–3745, December 2006.

[10] K. Engan, S. O. Aase, and J. H. Husoy. Frame based signal compression using method of optimal directions (MOD). In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1999.

[11] N. Jojic, B. Frey, and A. Kannan. Epitomic analysis of appearance and shape. In *Proc. ICCV*, 2003.

[12] K. Kavukcuoglu, M. Ranzato, R. Fergus, and Y. LeCun. Learning invariant features through topographic filter maps. In *Proc. CVPR*, 2009.

[13] B. Mailhé, S. Lesage, R. Gribonval, F. Bimbot, and P. Vandergheynst. Shift-invariant dictionary learning for sparse representations: extending K-SVD. In *Proc. EUSIPCO*, 2008.

[14] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11:19–60, 2010.

[15] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Non-local sparse models for image restoration. In *Proc. ICCV*, 2009.

[16] S. Mallat. *A Wavelet Tour of Signal Processing, Second Edition*. Academic Press, New York, 1999.

[17] S. Mallat and Z. Zhang. Matching pursuit in a time-frequency dictionary. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, 1993.

[18] Y. Nesterov. A method for solving the convex programming problem with convergence rate $O(1/k^2)$. *Dokl. Akad. Nauk SSSR*, 269:543–547, 1983.

[19] Y. Nesterov. Gradient methods for minimizing composite objective function. Technical report, Center for Operations Research and Econometrics (CORE), Catholic University of Louvain, 2007.

[20] K. Ni, A. Kannan, A. Criminisi, and J. Winn. Epitomic location recognition. In *Proc. CVPR*, 2008.

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

[22] G. Peyré. Sparse modeling of textures. *Journal of Mathematical Imaging and Vision*, 34(1):17–31, 2009.

[23] J. Thiagarajan, K. Ramamurthy, and A. Spanias. Shift-invariant sparse representation of images using learned dictionaries. In *IEEE International Workshop on Machine Learning for Signal Processing*, 2008.

[24] R. Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society. Series B*, 58(1):267–288, 1996.