Solving Inverse Problems with Piecewise Linear Estimators: From Gaussian Mixture Models to Structured Sparsity

Guoshen YU*[1], Guillermo SAPIRO[1], and Stéphane MALLAT[2]

[1]ECE, University of Minnesota, Minneapolis, Minnesota, 55414, USA
[2]CMAP, Ecole Polytechnique, 91128 Palaiseau Cedex, France

EDICS: TEC-RST

***Abstract*—A general framework for solving image inverse problems with piecewise linear estimations is introduced in this paper. The approach is based on Gaussian mixture models, estimated via a MAP-EM algorithm. A dual mathematical interpretation of the proposed framework with structured sparse estimation is described, which shows that the resulting piecewise linear estimate stabilizes the estimation when compared to traditional sparse inverse problem techniques. We demonstrate that in a number of image inverse problems, including interpolation, zooming, and deblurring of narrow kernels, the same simple and computationally efficient algorithm yields results in the same ballpark as the state-of-the-art.**

I. INTRODUCTION

Image restoration often requires to solve an inverse problem. It amounts to estimate an image $\mathbf{f}$ from a measurement

$$\mathbf{y} = \mathbf{U}\mathbf{f} + \mathbf{w},$$

obtained through a non-invertible linear degradation operator $\mathbf{U}$, and contaminated by an additive noise $\mathbf{w}$. Typical degradation operators include masking, subsampling in a uniform grid and convolution, the corresponding inverse problems often named interpolation, zooming and deblurring. Estimating $\mathbf{f}$ requires some prior information on the image, or equivalently image models. Finding good image models is therefore at the heart of image estimation.

Mixture models are often used as image priors since they enjoy the flexibility of signal description by assuming that the signals are generated by a mixture of probability distributions [57]. Gaussian mixture models (GMM) have been shown to provide powerful tools for data classification and segmentation applications (see for example [13], [32], [62], [68]), however, they have not yet been shown to generate state-of-the-art in a general class of image inverse problems, though very good initial steps were often reported. Ghahramani and Jordan have applied GMM for learning from incomplete data, i.e., images degraded by a masking operator, and have shown good classification results, however, it does not lead to state-of-the-art interpolation [33]. Portilla et al. have shown impressive image denoising results by assuming Gaussian scale mixture models (deviating from GMM by assuming different scale factors in the mixture of Gaussians) on wavelet representations [36], [49], [65], and have recently extended its applications on image deblurring [35]. Recently, Zhou et al. have developed an nonparametric Bayesian approach using more elaborated models, such as beta and Dirichlet processes, which leads to excellent results in denoising and interpolation [84].

The now popular sparse signal models, on the other hand, assume that the signals can be accurately represented with a few coefficients selecting atoms in some dictionary [53], [64]. Recently, very impressive image restoration results have been obtained with local patch-based sparse representations calculated with dictionaries learned from natural images [1], [24], [48], [51], [76]. Relative to pre-fixed dictionaries such as wavelets [53], curvelets [11], and bandlets [54], learned dictionaries enjoy the advantage of being better adapted to the images, thereby enhancing the sparsity. However, dictionary learning is a large-scale and highly non-convex problem. It requires high computational complexity, and its mathematical behavior is not yet well understood. In the dictionaries aforementioned, the actual sparse image representation is calculated with relatively expensive non-linear estimations, such as l_1 or matching pursuits [19], [23], [56]. More importantly, as will be reviewed in Section III-A, with a full degree of freedom in selecting the approximation space (atoms of the dictionary), non-linear sparse inverse problem estimation may be unstable and imprecise due to the coherence of the dictionary [55].

Structured sparse image representation models further regularize the sparse estimation by assuming dependency on the selection of the active atoms. One simultaneously selects blocks of approximation atoms, thereby reducing the number of possible approximation spaces [3], [26], [27], [39], [40], [69]. These structured approximations have been shown to improve the signal estimation in a compressive sensing context for a random operator $\mathbf{U}$. However, for more unstable inverse problems such as zooming or deblurring, this regularization by itself is not sufficient to reach state-of-the-art results. Recently some good image zooming results have been obtained with structured sparsity based on directional block structures in wavelet representations [55]. However, this directional regularization is not general enough to be extended to solve other inverse problems.

The Gaussian mixture models (GMM) developed in this work lead to piecewise linear estimators. [1] Image patches are far from Gaussian, neither are they necessarily mixture of Gaussians; on the other hand, piecewise linear approximations being

[1]The name "piecewise linear estimation" comes from the fact that for each Gaussian, the estimator is linear, and then a non-linearity appears in the selection of the best Gaussian model.

optimal for GMM, remain effective for much larger classes of functions and processes, including natural image patches as here demonstrated. Comparing with fully non-linear estimations based on the sparse models, piecewise linear estimations dramatically reduce the degree of freedom in the estimations, and are thus more stable. The piecewise linear estimations, calculated with a simple MAP-EM (maximum a posteriori expectation-maximization) algorithm, learns GMM from the degraded image, and yield results in the same ballpark as the state-of-the-art in a number of imaging inverse problems, often better than much more sophisticated algorithms based on more complex models, and at a lower computational cost.

The MAP-EM algorithm is described in Section II. After briefly reviewing sparse inverse problem estimation approaches, a mathematical equivalence between the proposed piecewise linear estimation (PLE) from GMM and structured sparse estimation is shown in Section III. This connection shows that PLE stabilizes the sparse estimation with a structured learned overcomplete dictionary composed of a union of PCA (Principal Component Analysis) bases, and with collaborative prior information incorporated in the eigenvalues, that privileges in the estimation the atoms that are more likely to be important. This interpretation suggests also an effective dictionary motivated initialization for the MAP-EM algorithm. In Section IV we support the importance of different components of the proposed PLE via some initial experiments. Applications of the proposed PLE in image interpolation, zooming, and deblurring are presented in sections V, VI, and VII respectively, and are compared with previous state-of-the-art methods. Conclusions are drawn in Section VIII.

II. Piecewise Linear Estimation

This section describes the Gaussian mixture models (GMM) and the MAP-EM algorithm, which lead to the proposed piecewise linear estimation (PLE).

A. Gaussian Mixture Models

Natural images include rich and non-stationary content, whereas when restricted to local windows, image structures appear to be simpler and are therefore easier to model. Following some previous works [1], [10], [51], an image is decomposed into overlapping $\sqrt{N}\times\sqrt{N}$, typically 8×8 following previous works [1], [24], local patches

$$\mathbf{y}_i = \mathbf{U}_i\mathbf{f}_i + \mathbf{w}_i, \tag{1}$$

where $\mathbf{U}_i$ is the degradation operator, for example random masking, subsampling or convolution, restricted to the patch i, $\mathbf{y}_i$, $\mathbf{f}_i$ and $\mathbf{w}_i$ are respectively the degraded, original image patches and the noise restricted to the patch, with $1\le i\le I$, I being the total number of patches. Treated as a signal, each of the patches is estimated, and their corresponding estimates are finally combined and averaged, leading to the estimate of the image. Note that for non-diagonal operator $\mathbf{U}_i$ such as blurring, special care needs to be taken for boundary issue, and the performance of the patch-based methods is generally limited as the size of the non-diagonal operator becomes large relative to the patch size. This will be further detailed in Section VII.

GMM describes local image patches with a mixture of Gaussian distributions. Assume there exist K Gaussian distributions $\{\mathcal{N}(\mu_k,\Sigma_k)\}_{1\le k\le K}$ parametrized by their means μ_k and covariances Σ_k. Each image patch $\mathbf{f}_i$ is independently drawn from one of these Gaussians with an unknown index $k_i\in[1,K]$, and with equal probability, whose probability density function is

$$p(\mathbf{f}_i) = \frac{1}{(2\pi)^{N/2}|\Sigma_{k_i}|^{1/2}}\exp\left(-\frac{1}{2}(\mathbf{f}_i-\mu_{k_i})^T\Sigma_{k_i}^{-1}(\mathbf{f}_i-\mu_{k_i})\right). \tag{2}$$

Estimating $\{\mathbf{f}_i\}_{1\le i\le I}$ from $\{\mathbf{y}_i\}_{1\le i\le I}$ can then be casted into the following problems:

- Estimate the Gaussian parameters $\{(\mu_k,\Sigma_k)\}_{1\le k\le K}$, from the degraded data $\{\mathbf{y}_i\}_{1\le i\le I}$.
- Identify the Gaussian distribution k_i that generates the patch i, $\forall 1\le i\le I$.
- Estimate $\mathbf{f}_i$ from its corresponding Gaussian distribution (μ_{k_i},Σ_{k_i}), $\forall 1\le i\le I$.

These problems are overall non-convex. The next section will present a maximum a posteriori expectation-maximization (MAP-EM) algorithm that calculates a local-minimum solution [2].

B. MAP-EM Algorithm

Following an initialization, addressed in Section III-C, the MAP-EM algorithm is an iterative procedure that alternates between two steps. In the E-step, assuming that the estimates of the Gaussian parameters $\{(\tilde{\mu}_k,\tilde{\Sigma}_k)\}_{1\le k\le K}$ are known (following the previous M-step), for each patch one calculates the maximum a posteriori (MAP) estimates $\tilde{\mathbf{f}}_i^k$ with all the Gaussian models, and selects the best Gaussian model $\tilde{k}_i$ to obtain the estimate of the patch $\tilde{\mathbf{f}}_i=\tilde{\mathbf{f}}_i^{\tilde{k}_i}$. In the M-step, assuming that the Gaussian model selection $\tilde{k}_i$ and the signal estimate $\tilde{\mathbf{f}}_i$, $\forall i$, are known (following the previous E-step), one estimates (updates) the Gaussian models $\{(\tilde{\mu}_k,\tilde{\Sigma}_k)\}_{1\le k\le K}$.

1) E-step: Signal Estimation and Model Selection: In the E-step, the estimates of the Gaussian parameters $\{(\tilde{\mu}_k,\tilde{\Sigma}_k)\}_{1\le k\le K}$ are assumed to be known. To simplify the notation, we assume without loss of generality that the Gaussians have zero means $\tilde{\mu}_k=\mathbf{0}$, as one can always center the image patches with respect to the means.

For each image patch i, the signal estimation and model selection is calculated to maximize the log a-posteriori probability $\log p(\mathbf{f}_i|\mathbf{y}_i,\tilde{\Sigma}_{k_i})$:

$$\begin{aligned}(\tilde{\mathbf{f}}_i,\tilde{k}_i) &= \arg\max_{\mathbf{f},k}\log p(\mathbf{f}|\mathbf{y}_i,\tilde{\Sigma}_k) = \arg\max_{\mathbf{f},k}\left(\log p(\mathbf{y}_i|\mathbf{f},\tilde{\Sigma}_k)+\log p(\mathbf{f}|\tilde{\Sigma}_k)\right)\\ &= \arg\min_{\mathbf{f},k}\left(\|\mathbf{U}_i\mathbf{f}-\mathbf{y}_i\|^2+\sigma^2\mathbf{f}^T\tilde{\Sigma}_k^{-1}\mathbf{f}+\sigma^2\log|\tilde{\Sigma}_k|\right),\end{aligned} \tag{3}$$

where the second equality follows the Bayes rule and the third one is derived with the assumption that $\mathbf{w}_i\sim\mathcal{N}(\mathbf{0},\sigma^2 Id)$, with Id the identity matrix, and $\mathbf{f}\sim\mathcal{N}(\mathbf{0},\tilde{\Sigma}_k)$.

The maximization is first calculated over $\mathbf{f}$ and then over k. Given a Gaussian signal model $\mathbf{f}\sim\mathcal{N}(\mathbf{0},\tilde{\Sigma}_k)$, it is well known that the MAP estimate

$$\tilde{\mathbf{f}}_i^k = \arg\min_{\mathbf{f}}\left(\|\mathbf{U}_i\mathbf{f}-\mathbf{y}_i\|^2+\sigma^2\mathbf{f}^T\tilde{\Sigma}_k^{-1}\mathbf{f}\right) \tag{4}$$

minimizes the risk $E[\|\tilde{\mathbf{f}}_i^k-\mathbf{f}_i\|^2]$ [53]. One can verify that the solution to (4) can be calculated with a linear filtering

$$\tilde{\mathbf{f}}_i^k = \mathbf{W}_{k,i}\mathbf{y}_i, \tag{5}$$

where

$$\mathbf{W}_{k,i} = \Sigma_k\mathbf{U}_i^T(\mathbf{U}_i\Sigma_k\mathbf{U}_i^T+\sigma^2 Id)^{-1} \tag{6}$$

is a Wiener filter matrix. Since $\mathbf{U}_i\Sigma_k\mathbf{U}_i^T$ is semi-positive definite, $\mathbf{U}_i\Sigma_k\mathbf{U}_i^T+\sigma^2 Id$ is positive definite and its inverse is well defined.

The best Gaussian model $\tilde{k}_i$ that generates the maximum MAP probability among all the models is then selected with the estimated $\tilde{\mathbf{f}}_i^k$

$$\tilde{k}_i = \arg\min_k \left(\|\mathbf{U}_i\tilde{\mathbf{f}}_i^k - \mathbf{y}\|^2 + \sigma^2 (\tilde{\mathbf{f}}_i^k)^T \Sigma_k^{-1} \tilde{\mathbf{f}}_i^k + \sigma^2 \log\left|\tilde{\Sigma}_k\right| \right). \quad (7)$$

The signal estimate is obtained by plugging in the best model $\tilde{k}_i$ in the MAP estimate (4)

$$\tilde{\mathbf{f}}_i = \tilde{\mathbf{f}}_i^{\tilde{k}_i}. \quad (8)$$

The whole E-step is basically calculated with a set of linear filters. For typical applications such as zooming and deblurring where the degradation operators $\mathbf{U}_i$ are translation-invariant and do not depend on the patch index i, i.e., $\mathbf{U}_i \equiv \mathbf{U}$, the Wiener filter matrices $\mathbf{W}_{k,i} \equiv \mathbf{W}_k$ (6) can be precomputed for the K Gaussian distributions. Calculating (5) thus requires only $2N^2$ floating-point operations (flops), where N is the image patch size. For a translation-variant degradation $\mathbf{U}_i$, random masking for example, $\mathbf{W}_{k,i}$ needs to be calculated at each position where $\mathbf{U}_i$ changes. Since $\mathbf{U}_i\Sigma_k\mathbf{U}_i^T+\sigma^2 Id$ is positive definite, the matrix inversion can be implemented with $N^3/3+2N^2 \approx N^3/3$ flops through a Cholesky factorization [9]. All this makes the E-step computationally efficient.

Note that in the case $\mathbf{U}_i$ is a masking or subsampling operator, which maps from $\mathbb{R}^N$ to $\mathbb{R}^{\frac{N}{S}}$ extracting N/S entries of $\mathbf{f}_i \in \mathbb{R}^N$, where S is the masking or subsampling ratio, $\mathbf{U}_i$ can be written as a matrix of size $\frac{N}{S}\times N$ by removing the zero rows, and $\mathbf{y}_i = \mathbf{U}_i\mathbf{f}_i + \mathbf{w}_i$ can be written in $\mathbb{R}^{\frac{N}{S}}$. The matrix inversion in (6) thus involves a matrix of size $\frac{N}{S}\times\frac{N}{S}$ instead of $N\times N$, further considerably reducing the computational complexity of the E-step from $N^3/3$ to $\frac{N^3}{3S^3}$, as the $\mathbf{U}_i$ is translation-variant.

2) M-step: Model Estimation: In the M-step, the Gaussian model selection $\tilde{k}_i$ and the signal estimate $\tilde{\mathbf{f}}_i$ of all the patches are assumed to be known. Let $\mathscr{C}_k$ be the ensemble of the patch indices i that are assigned to the k-th Gaussian model, i.e., $\mathscr{C}_k = \{i : \tilde{k}_i = k\}$, and let $|\mathscr{C}_k|$ be its cardinality. The parameters of each Gaussian model are estimated with the maximum likelihood (ML) estimate using all the patches assigned to that Gaussian cluster,

$$(\tilde{\mu}_k, \tilde{\Sigma}_k) = \arg\max_{\mu_k,\Sigma_k} \log p(\{\tilde{\mathbf{f}}_i\}_{i\in\mathscr{C}_k} | \mu_k, \Sigma_k). \quad (9)$$

With the Gaussian model (2) , one can easily verify that the resulting estimate is the empirical estimate

$$\tilde{\mu}_k = \frac{1}{|\mathscr{C}_k|}\sum_{i\in\mathscr{C}_k}\tilde{\mathbf{f}}_i \quad \text{and} \quad \tilde{\Sigma}_k = \frac{1}{|\mathscr{C}_k|}\sum_{i\in\mathscr{C}_k}(\tilde{\mathbf{f}}_i-\tilde{\mu}_k)(\tilde{\mathbf{f}}_i-\tilde{\mu}_k)^T. \quad (10)$$

The empirical covariance estimate may be improved through regularization when there is lack of data [67] (for typical patch size 8×8, the dimension of the covariance matrix Σ_k is 64×64, while the $|\mathscr{C}_k|$ is typically in the order of a few hundred). A simple and standard eigenvalue-based regularization [2] is used here

$$\tilde{\Sigma}_k \leftarrow \tilde{\Sigma}_k + \varepsilon Id, \quad (11)$$

where ε is a small constant. The regularization also guarantees that the estimate $\tilde{\Sigma}_k$ of the covariance matrix is full-rank, which stabilizes the covariance matrix inversion, and is important for the Gaussian model selection (7), since if $\tilde{\Sigma}_k$ is not full rank, then $\log\left|\tilde{\Sigma}_k\right| \to -\infty$, biasing the model selection. The computational complexity of the M-step is negligible with respect to the E-step.

As the MAP-EM algorithm described above iterates, the MAP probability of the observed signals $p(\{\tilde{\mathbf{f}}_i\}_{1\le i\le I}|\{\mathbf{y}_i\}_{1\le i\le I}, \{\tilde{\mu}_k, \tilde{\Sigma}_k\}_{1\le k\le K})$ always increases. This can be observed by interpreting the E- and M-steps as a coordinate descent optimization [38]. In the experiments, the convergence of the patch clustering and resulting PSNR is always observed.

Note that we name the above algorithm MAP-EM as its two steps go in parallel with those of the classic EM algorithm [21] applied to the point clustering problem under the GMM [16], with an extra MAP estimate in our E-step, as the original signals are not observed but need to be estimated from the degraded observations. The algorithm is also interpretable as an instance of the greedy Iterated Conditional Modes (ICM) algorithm [5].

The MAP-EM implements a piecewise linear estimation, as it estimates a piecewise Gaussian model from the image patches, and for each image patch selects one best fit Gaussian model and estimates the signal with the linear estimation therein.

III. PLE and Structured Sparse Estimation

The MAP-EM algorithm described above requires an initialization. A good initialization is highly important for iterative algorithms that try to solve non-convex problems, and remains an active research topic [4], [31]. This section describes a dual structured sparse interpretation of GMM and MAP-EM, which suggests an effective dictionary motivated initialization for the MAP-EM algorithm. Moreover, it shows that the resulting piecewise linear estimate stabilizes traditional sparse inverse problem estimation.

The sparse inverse problem estimation approaches will be first reviewed. After describing the connection between MAP-EM and structured sparsity via estimation in PCA bases, an intuitive and effective initialization will be presented.

A. Sparse Inverse Problem Estimation

Traditional sparse super-resolution estimation in dictionaries provides effective non-parametric approaches to inverse problems, although the coherence of the dictionary and their large degree of freedom may become sources of instability and errors. [2]These algorithms are briefly reviewed in this section. "Super-resolution" is loosely used here as these approaches try to recover information that is lost after the degradation.

A signal $\mathbf{f}\in\mathbb{R}^N$ is estimated by taking advantage of prior information which specifies a dictionary $\mathbf{D}\in\mathbf{R}^{N\times|\Gamma|}$, having $|\Gamma|$ columns corresponding to atoms $\{\phi_m\}_{m\in\Gamma}$, where $\mathbf{f}$ has a sparse approximation. This dictionary may be a basis or some redundant frame, with $|\Gamma|\ge N$. Sparsity means that $\mathbf{f}$ is well approximated by its orthogonal projection $\mathbf{f}_\Lambda$ over a subspace $\mathbf{V}_\Lambda$ generated by a small number $|\Lambda| \ll |\Gamma|$ of column vectors $\{\phi_m\}_{m\in\Lambda}$ of $\mathbf{D}$:

$$\mathbf{f} = \mathbf{f}_\Lambda + \varepsilon_\Lambda = \mathbf{D}(\mathbf{a}\cdot\mathbf{1}_\Lambda) + \varepsilon_\Lambda, \quad (12)$$

where $\mathbf{a}\in\mathbb{R}^{|\Gamma|}$ is the transform coefficient vector, $\mathbf{a}\cdot\mathbf{1}_\Lambda$ selects the coefficients in Λ and sets the others to zero, $\mathbf{D}(\mathbf{a}\cdot\mathbf{1}_\Lambda)$

[2]While in some context "super-resolution" is referred to as the approaches that calculate a high-resolution image from observed multiple low-resolution ones, in this paper "super-resolution" means recovering a whole signal from partial measurements [53].

multiplies the matrix $\mathbf{D}$ with the vector $\mathbf{a}\cdot\mathbf{1}_\Lambda$, and $\|\varepsilon_\Lambda\|^2 \ll \|\mathbf{f}\|^2$ is a small approximation error.

Sparse inversion algorithms try to estimate from the degraded signal $\mathbf{y}=\mathbf{U}\mathbf{f}+\mathbf{w}$ the support Λ and the coefficients $\mathbf{a}$ in Λ that specify the projection of $\mathbf{f}$ in the approximation space $\mathbf{V}_\Lambda$. It results from (12) that

$$\mathbf{y}=\mathbf{U}\mathbf{D}(\mathbf{a}\cdot\mathbf{1}_\Lambda)+\varepsilon', \quad \text{with} \quad \varepsilon'=\mathbf{U}\varepsilon+\mathbf{w}. \tag{13}$$

This means that $\mathbf{y}$ is well approximated by the same sparse set Λ of atoms and the same coefficients $\mathbf{a}$ in the transformed dictionary $\mathbf{UD}$, whose columns are the transformed vectors $\{\mathbf{U}\phi_m\}_{m\in\Gamma}$.

Since $\mathbf{U}$ is not an invertible operator, the transformed dictionary $\mathbf{UD}$ is redundant, with column vectors which are linearly dependent. It results that $\mathbf{y}$ has an infinite number of possible decompositions in $\mathbf{UD}$. A sparse approximation $\tilde{\mathbf{y}}=\mathbf{U}\mathbf{D}\tilde{\mathbf{a}}$ of $\mathbf{y}$ can be calculated with a basis pursuit algorithm which minimizes a Lagrangian penalized by a sparse l_1 norm [15], [71]

$$\tilde{\mathbf{a}}=\arg\min_{\mathbf{a}} \|\mathbf{U}\mathbf{D}\mathbf{a}-\mathbf{y}\|^2+\lambda\,\|\mathbf{a}\|_1, \tag{14}$$

or with faster greedy matching pursuit algorithms [56]. The resulting sparse estimation of $\mathbf{f}$ is

$$\tilde{\mathbf{f}}=\mathbf{D}\tilde{\mathbf{a}}. \tag{15}$$

As we explain next, this simple approach is not straightforward and often not as effective as it seems. The *Restrictive Isometry Property* of Candès and Tao [12] and Donoho [22] is a strong sufficient condition which guarantees the correctness of the penalized l_1 estimation. This restrictive isometry property is valid for certain classes of operators $\mathbf{U}$, but not for important structured operators such as subsampling on a uniform grid or convolution. For structured operators, the precision and stability of this sparse inverse estimation depends upon the "geometry" of the approximation support Λ of $\mathbf{f}$, which is not well understood mathematically, despite some sufficient exact recovery conditions proved for example by Tropp [72], and many others (mostly related to the coherence of the equivalent dictionary). Nevertheless, some necessary qualitative conditions for a precise and stable sparse super-resolution estimate (15) can be deduced as follows [53], [55]:

- **Sparsity. D** provides a sparse representation for $\mathbf{f}$.
- **Recoverability.** The atoms have non negligible norms $\|\mathbf{U}\phi_m\|^2 \gg 0$. If the degradation operator $\mathbf{U}$ applied to ϕ_m leaves no "trace," the corresponding coefficient $\mathbf{a}[m]$ can not be recovered from $\mathbf{y}$ with (14). We will see in the next subsection that this recoverability property of transformed relevant atoms having sufficient energy is critical for the GMM/MAP-EM introduced in the previous section as well.
- **Stability.** The transformed dictionary $\mathbf{UD}$ is incoherent enough. Sparse inverse problem estimation may be unstable if some columns $\{\mathbf{U}\phi_m\}_{m\in\Gamma}$ in $\mathbf{UD}$ are too similar. To see this, let us imagine a toy example, where a constant-value atom and a highly oscillatory atom (with values $-1,1,-1,1,\ldots$), after a $\times 2$ subsampling, become identical. The sparse estimation (14) can not distinguish between them, which results in an unstable inverse problem estimate (15). The coherence of $\mathbf{UD}$ depends on $\mathbf{D}$ as well as on the operator $\mathbf{U}$. Regular operators $\mathbf{U}$ such as subsampling on a uniform grid and convolution, usually lead to a coherent $\mathbf{UD}$, which makes accurate inverse problem estimation difficult.

Several authors have applied this sparse super-resolution framework (14) and (15) for image inverse problems. Sparse estimation in dictionaries of curvelet frames and DCT have been applied successfully to interpolation of randomly sampled images [25], [29], [37]. However, for uniform grid interpolations, Section VI shows that the resulting interpolation estimations are not as precise as simple linear bicubic interpolations. A contourlet zooming algorithm [59] can provide a slightly better PSNR than a bicubic interpolation, but the results are considerably below the state-of-the-art. Learned dictionaries of image patches have generated good interpolation results [51], [84]. In some recent works sparse super-resolution algorithms with learned dictionary have been studied for zooming and deblurring [48], [76]. As shown in sections VI and VII, although they sometimes produce good visual quality, they often generate artifacts and the resulting PSNRs are not as good as more standard methods.

Another source of instability of these algorithms comes from their full degree of freedom. The non-linear approximation space $\mathbf{V}_\Lambda$ is estimated by selecting the approximation support Λ, with basically no constraint. A selection of $|\Lambda|$ atoms from a dictionary of size $|\Gamma|$ thus corresponds to a choice of an approximation space among $\binom{|\Gamma|}{|\Lambda|}$ possible subspaces. In a local patch-based sparse estimation with 8×8 patch size, typical values of $|\Gamma|=256$ and $|\Lambda|=8$ lead to a huge degree of freedom $\binom{256}{8}\sim 10^{14}$, further stressing the inaccuracy of estimating $\mathbf{a}$ from an $\mathbf{UD}$.

These issues are addressed with the proposed PLE framework and its mathematical connection with structured sparse models described next.

B. Structured Sparse Estimation in PCA bases

The PCA bases bridge the GMM/MAP-EM framework presented in Section II with the sparse estimation described above. For signals $\{\mathbf{f}_i\}$ following a statistical distribution, a PCA basis is defined as the matrix that diagonalizes the data covariance matrix $\Sigma_k=E[\mathbf{f}_i\mathbf{f}_i^T]$,

$$\Sigma_k=\mathbf{B}_k\mathbf{S}_k\mathbf{B}_k^T, \tag{16}$$

where $\mathbf{B}_k$ is the PCA basis and $\mathbf{S}_k=\mathrm{diag}(\lambda_1^k,\ldots,\lambda_N^k)$ is a diagonal matrix, whose diagonal elements $\lambda_1^k\geq\lambda_2^k\geq\ldots\geq\lambda_N^k$ are the sorted eigenvalues. It can be shown that the PCA basis is orthonormal, i.e., $\mathbf{B}_k^T\mathbf{B}_k=Id$, each of its columns ϕ_m^k, $1\leq m\leq N$, being an atom that represents one principal direction. The eigenvalues are non-negative, $\lambda_m\geq 0$, and measure the energy of the signals $\{\mathbf{f}_i\}$ in each of the principal directions [53].

Transforming $\mathbf{f}_i$ from the canonical basis to the PCA basis $\mathbf{a}_i^k=\mathbf{B}_k^T\mathbf{f}_i$, one can verify that the MAP estimate (4)-(6) can be equivalently calculated as

$$\tilde{\mathbf{f}}_i^k=\mathbf{B}_k\tilde{\mathbf{a}}_i^k, \tag{17}$$

where, following simple algebra and calculus, the MAP estimate of the PCA coefficients $\tilde{\mathbf{a}}_i^k$ is obtained by

$$\tilde{\mathbf{a}}_i^k=\arg\min_{\mathbf{a}}\left(\|\mathbf{U}_i\mathbf{B}_k\mathbf{a}-\mathbf{y}_i\|^2+\sigma^2\sum_{m=1}^{N}\frac{|\mathbf{a}[m]|^2}{\lambda_m^k}\right). \tag{18}$$

Comparing (18) with (14), the MAP-EM estimation can thus be interpreted as a structured sparse estimation. As illustrated in

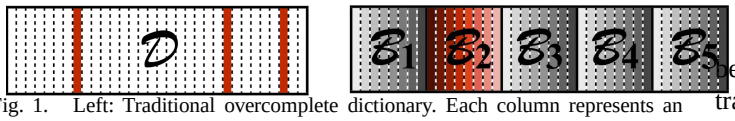


Fig. 1. Left: Traditional overcomplete dictionary. Each column represents an atom in the dictionary. Non-linear estimation has the full degree of freedom to select any combination of atoms (marked by the columns in red). Right: The underlying structured sparse piecewise linear dictionary of the proposed approach. The dictionary is composed of a family of PCA bases whose atoms are pre-ordered by their associated eigenvalues. For each image patch, an optimal *linear* estimator is calculated in each PCA basis and the best linear estimate among the bases is selected (marked by the basis in red).

Figure 1, the proposed dictionary has the advantage of the traditional learned overcomplete dictionaries being overcomplete, and adapted to the image under test thanks to the Gaussian model estimation in the M-step (which is equivalent to updating the PCAs), but the resulting piecewise linear estimator (PLE) is more structured than the traditional nonlinear sparse estimation. PLE is calculated with a *linear* estimation in each basis and a *non-linear* best basis selection:

- **Nonlinear block sparsity.** The dictionary is composed of a union of K PCA bases. To represent an image patch, the *non-linear* model selection (3) in the E-step restricts the estimation to only one basis (N atoms out of KN selected in group), and has a degree of freedom equal to K, sharply reduced from that in the traditional sparse estimation which has the full degree of freedom in atom selection.
- **Linear collaborative filtering.** Inside each PCA basis, the atoms are pre-ordered by their associated eigenvalues (which decay very fast as we will later see, leading to sparsity inside the block as well). In contrast to the non-linear sparse l_1 estimation (14), the MAP estimate (18) implements the regularization with the l_2 norm of the coefficients weighted by the eigenvalues $\{\lambda_m^k\}_{1\leq m\leq N}$, and is calculated with a *linear* filtering (5) (6). The eigenvalues are computed from all the signals $\{\mathbf{f}_i\}$ in the same Gaussian distribution class. The resulting estimation therefore implements a collaborative filtering which incorporates the information from all the signals in the same cluster. The weighting scheme privileges the coefficients $\mathbf{a}_i[m]$ corresponding to the principal directions with large eigenvalues λ_m, where the energy is likely to be high, and penalizes the others. For the ill-posed inverse problems, the collaborative prior information incorporated in the eigenvalues $\{\lambda_m^k\}_{1\leq m\leq N}$ further stabilizes the estimate.

  Note that this collaborative weighting is fundamentally different than the standard one used in iterative weighted l_2 approaches to sparse coding [20]. This collaborative filtering is also fundamentally different than the "collaborative Wiener filtering" in [17], both in signal modeling (GMM in this work and the nonlocal self-similarity models in [17]), and in patch clustering and signal estimation (in this work the patch clustering and signal estimation are jointly calculated by maximizing a MAP probability (3), which is optimal under the GMM model, whereas in [17] they are calculated respectively by the block matching and the empirical Wiener filtering). The collaboration in [17] follows from the spectral representation for the whole cluster, while here is obtained via the eigenvalues of the cluster's PCA.

Note that although PLE can be interpreted and connected with structured sparse modeling via PCA, the algorithm can be implemented as described in Section II without the PCA transform. As described in Section II, the complexity of the MAP-EM algorithm is dominated by the E-step. For an image patch size of $\sqrt{N}\times\sqrt{N}$ (typical value 8×8), it costs $2KN^2$ flops for translation-invariant degradation operators such as uniform subsampling and convolution, and $KN^3/3$ flops for translation-variant operators such as random masking, where K is the number of PCA bases. The overall complexity is therefore tightly upper bounded by $\mathcal{O}(2LKN^2)$ or $\mathcal{O}(LKN^3/3)$, where L is the number of iterations. As will be shown in Section IV, the algorithm converges fast for image inverse problems, typically in $L=3$ to 5 iterations. On the other hand, the complexity of the l_1 minimization with the same dictionary is $\mathcal{O}(KN^3)$, with typically a large factor in front as the l_1 converges slowly in practice. The MAP-EM algorithm is thus typically one or two orders of magnitude faster than the sparse estimation.

To conclude, let as come back to the recoverability property mentioned in the previous section. We see from (18) that if an eigenvector of the covariance matrix is killed by the operator $\mathbf{U}_i$, then its contribution to the recovery of $\mathbf{y}_i$ is virtually null, while it pays a price proportional to the corresponding eigenvalue. Then, it will not be used in the optimization (18), and thereby in the reconstruction of the signal following (17). This means that the wrong model might be selected and an inaccurate reconstruction obtained. This further stresses the importance of a correct design of dictionary elements, which from the description just presented, it is equivalent to the correct design of the covariance matrix, including the initialization, which is described next.

C. Initialization of MAP-EM

The PCA formulation just described not only reveals the connection between PLE and structured sparse estimations, but it is crucial for understanding how to initialize the Gaussian models for MAP-EM as well.

1) Sparsity: As explained in Section III-A, for the sparse inverse problem estimations to have the super-resolution ability, the first requirement on the dictionary is to be able to provide sparse representations of the image. It has been shown that capturing image directional regularity is highly important for sparse representations [1], [11], [54]. In dictionary learning, for example, most prominent atoms look like local edges good at representing contours, as illustrated in Figure 2-(a). Therefore the initial PCAs in our framework, which following (16) will lead to the initial Gaussians, are designed to capture image directional regularity.

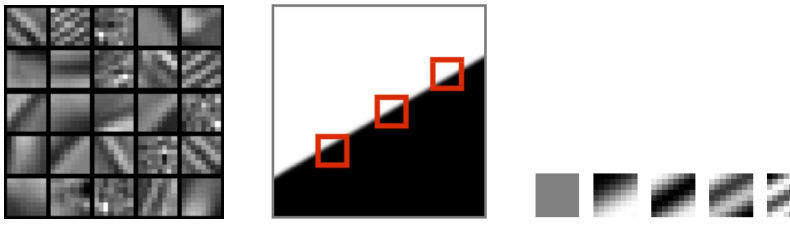


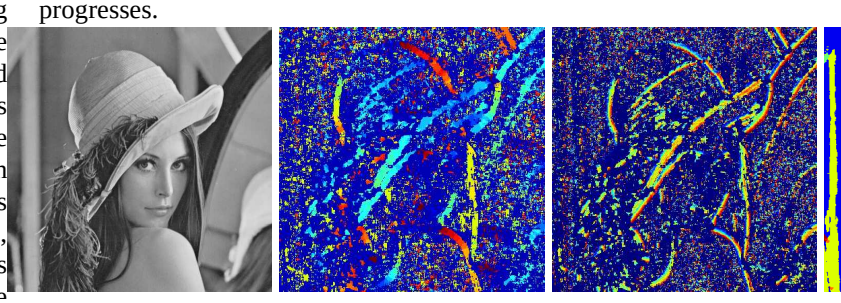
Fig. 2. (a) Some typical dictionary atoms learned from the image Lena (Figure 3-(a)) with K-SVD [1]. (b)-(d) A numerical procedure to obtain the initial directional PCAs. (b) A synthetic edge image. Patches (8×8) that touch the edge are used to calculate an initial PCA basis. (c) The first 8 atoms in the PCA basis with the largest eigenvalues. (d) Typical eigenvalues.

The initial directional PCA bases are calculated following a simple numerical procedure. Directions from 0 to π are uniformly sampled to K angles, and one PCA basis is calculated per angle. The calculation of the PCA at an angle θ uses a synthetic blank-and-white edge image following the same direction, as illustrated in Figure 2-(b). Local patches that touch the contour are collected and are used to calculate the PCA basis (following (10) and (16)). The first atom, which is almost DC, is replaced by DC, and a Gram-Schmidt orthogonalization is calculated on the other atoms to ensure the orthogonality of the basis. The patches contain edges that are translation-invariant. As the covariance of a stationary process is diagonalized by the Fourier basis, unsurprisingly, the resulting PCA basis has first few important atoms similar to the cosines atoms oscillating in the direction θ from low-frequency to high-frequency, as shown in Figure 2-(c). Comparing with the Fourier vectors, these PCAs enjoy the advantage of being free of the periodic boundary issue, so that they can provide sparse representations for local image patches. The eigenvalues of all the bases are initiated with the same ones obtained from the synthetic contour image, that have fast decay, Figure 2-(d). These, following (16), complete the covariance initialization. The Gaussian means are initialized with zeros.

It is worth noting that this directional PCA basis not only provides sparse representations for contours and edges, but it captures well textures of the same directionality as well. Indeed, in a space of dimension N corresponding to patches of size $\sqrt{N}\times\sqrt{N}$, the first about $\sqrt{N}$ atoms illustrated in Figure 2-(c) absorb most of the energy in local patterns following the same direction in real images, as indicated by the fast decay of the eigenvalues (very similar to Figure 2-(d)).

A typical patch size is $\sqrt{N}\times\sqrt{N}=8\times 8$, as selected in previous works [1], [24]. The number of directions in a local patch is limited due to the pixelization. The DCT basis is also included in competition with the directional bases to capture isotropic image patterns. Our experiments have shown that in image inverse problems, there is a significant average gain in PSNR when K grows from 0 to 3 (when $K=0$, the dictionary is initialized with only a DCT basis and all the patches are assigned to the same cluster), which shows that one Gaussian model, or equivalently a single linear estimator, is not enough to accurately describe the image. When K increases, the gain reduces and gets stabilized at about $K=36$. Compromising between performance and complexity, $K=18$, which corresponds to a 10° angle sampling step, is selected in all the future experiments.

Figures 3-(a) and (b) illustrates the Lena image and the corresponding patch clustering, i.e., the model selection $\tilde{k}_i$, obtained for the above initialization, calculated with (7) in the E-step described in Section II. The patches are densely overlapped and each pixel in Figure 3-(b) represents the model k_i selected for the 8×8 patch around it, different colors encoding different values of $\tilde{k}_i$, from 1 to 19 (18 directions plus a DCT). One can observe, for example on the edges of the hat, that patches where the image patterns follow similar directions are clustered together, as expected. Let us note that on the uniform regions such as the background, where there is no directional preference, all the bases provide equally sparse representations. As the $\log|\Sigma_k|=\Pi_{m=1}^{N}\lambda_m^k$ term in the model selection (7) is initialized as identical for all the Gaussian models, the clustering is random inf these regions. The clustering will improve as the MAP-EM progresses.



(a) **(b)** **(c)**

Fig. 3. (a). Lena image. ((b) to (d) are color images.) (b). Patch clustering obtained with the initial directional PCAs (see Figure 2-(c)). The patches are densely overlapped and each pixel represents the model $\tilde{k}_i$ selected for the 8×8 patch around it, different colors encoding different direction values of $\tilde{k}_i$, from 1 to $K=19$. (c). Patch clustering obtained with the initial position PCAs (see Figure 4). Different colors encoding different position values of k_i, from 1 to $P=12$. (d) and (e). Patch clustering with respectively directional and position PCAs after the 2nd iteration.

2) Recoverability: The oscillatory atoms illustrated in Figure 2-(c) are spread in space. Therefore, for diagonal operators in space such as masking and subsampling, they satisfy well the recoverability condition $\|\mathbf{U}\phi_m^k\|^2\geq 0$ for super-resolution described in Section III-A. However, as these oscillatory atoms have Dirac supports in Fourier, for convolution operators, the recoverability condition is violated. For convolution operators $\mathbf{U}$, $\|\mathbf{U}\phi_m^k\|^2\geq 0$ requires that the atoms have spread Fourier spectrum. Spatially localized atoms have spread Fourier spectrum. Following a similar numerical scheme as described above, patches touching the edge at a *fixed* position are extracted from synthetic edge images with different amounts of blur. The resulting PCA basis, named position PCA basis hereafter, contains localized atoms of different polarities and at different scales, following the same direction θ, as illustrated in Figure 4 (which look like wavelets along the appropriate direction). For each direction θ, a family of localized PCA bases $\{\mathbf{B}_{k,p}\}_{1\leq p\leq P}$ are calculated at all the positions translating within the patch. The eigenvalues are initialized with the same fast decay ones as illustrated in Figure 2-(d) for all the position PCA bases. Each pixel in Figure 3-(c) represents the model p_i selected for the 8×8 patch around it, different colors encoding different position values of p_i, from 1 to 12. The rainbow-like color transitions on the edges show that the position bases are accurately fitted to the image structures. Note that although the position PCA bases consisting of localized atoms may provide more sparse representation for localized edges, as opposed to the directional PCA bases they do not satisfy the recoverability condition under masking degradation operators, and are thus less appropriate for solving interpolation problems.

A summary of the complete algorithm is given in Figure 5. The MAP-EM algorithm, with an imaging-motivated initialization, leads to successful applications in a number of image inverse problems as will be shown below.

Fig. 4. The first 8 atoms in the position PCA basis with the largest eigenvalues.

3) Wiener Filtering Interpretation: Figure 6 illustrates some typical Wiener filters, which are the rows of $\mathbf{W}_k$ in (6), calculated with the initial PCA bases described above for zooming and deblurring. The filters have intuitive interpretations, for example directional interpolator for zooming and directional deconvolution for deblurring, confirming the effectiveness of the initialization.

The MAP-EM algorithm of the PLE image inverse problem estimate

1) **Initialization.**
   - Eigenvectors. The initial eigenvectors of each Gaussian are calculated following the numerical procedure detailed in Section III-C1 for inpainting and zooming, and sections III-C2 and VII-A for deblurring.
   - Eigenvalues. The initial eigenvalues are obtained by calculating the eigenvalues of a collection of natural image patches. These same initial eigenvalues are used for all the Gaussians.

   The eigenvectors and eigenvalues are calculated once and stored. For each experiment they are then loaded.
2) **E-step.** For each image patch, estimate the original signal and its Gaussian identity by (5), (7) and (8).
3) **M-step.** For each Gaussian, estimate its mean and covariance matrix by (10) and (11).
4) If not converged, go to Step 2. (The algorithm typically converges in 3 to 5 iterations.)

Note that while the PCA formulation (18) *reveals a connection between PLE and structured sparse models, the algorithm is actually implemented without PCA.*

Fig. 5 of the algorithm

(a) **(b)** **(c)** **(d)**

Fig. 6. Some filters generated by the MAP estimator. (a) and (b) are for image zooming, where the degradation operator $\mathbf{U}$ is a 2×2 subsampling operator. Gray-level from white to black: values from negative to positive. (a) is computed with a Gaussian distribution whose PCA basis is a DCT basis, and it implements an isotropic interpolator. (b) is computed with a Gaussian distribution whose PCA basis is a directional PCA basis (angle $\theta = 30°$), and it implements a directional interpolator. (c) and (d) are shown in Fourier and are for image deblurring, where the degradation operator $\mathbf{U}$ is a Gaussian convolution operator. Gray-level from white to black: Fourier modules from zero to positive. (c) is computed with a Gaussian distribution whose PCA basis is a DCT basis, and it implements an isotropic deblurring filter. (d) is computed with a Gaussian distribution whose PCA basis is a directional PCA basis (angle $\theta = 30°$, at a fixed position), and it implements a directional deblurring filter.

D. Additional Comments on Related Works

Before proceeding with experimental results and applications, let us further comment on some related works, in addition to those already addressed in Section I.

The MAP-EM algorithm using various probability distributions such as Gaussian, Laplacian, Gamma and Gibbs, has been widely applied in medical image reconstruction and analysis (see for example [83], [47]). Following the Gaussian mixture models, MAP-EM alternates between image patch estimation and clustering, and Gaussian models estimation.

Clustering-based estimation based on self-similarity image models has been shown effective for image restoration [10], [17], [41], [45], [50], [63]. In these works, similar patches are clustered typically using the block matching technique, i.e., including in the same cluster the patches among which the Euclidian distance or mean absolute difference is small. Image segmentation algorithms such as k-means on local image features has been considered as well [14]. While such clustering is intuitive, the clustering and signal estimation are addressed as two separate problems [14], [50]. The self-similarity patch-based approaches have been equally addressed in the framework of partial differential equations [28], [34], [73]. The generalized PCA [75] models and segments data using an algebraic subspace clustering technique based on polynomial fitting and differentiation, and while it has been shown effective in image segmentation, it does not reach state-of-the-art in image restoration. In the recent non-parametric Bayesian approach [84], an image patch clustering is implemented with probability models, which improves the denoising and interpolation results, although still under performing, in quality and computational cost, the framework here introduced.

Based on the Gaussian mixture models here developed, the clustering in this framework is calculated jointly with the signal estimation as one consistent problem by maximizing the MAP probability (3). The effectiveness of this modeling will be further supported next with examples in a number of imaging inverse problem applications.

IV. Initial Supportive Experiments

Before proceeding with detailed experimental results for a number of applications of the proposed framework, this section shows through some basic experiments the effectiveness and importance of the initialization proposed above, the evolution of the representations as the MAP-EM algorithm iterates, as well as the improvement brought by the structure in PLE with respect to traditional sparse estimation.

Following some recent works, e.g., [52], an image is decomposed into 128×128 regions, each region treated with the MAP-EM algorithm separately. The idea is that image contents are often more coherent semi-locally than globally, and Gaussian model estimation or dictionary learning can be slightly improved in semi-local regions. This also saves memory and enables the processing to proceed as the image is being transmitted. Parallel processing on image regions is also possible when the whole image is available. Regions are half-overlapped to eliminate the boundary effect between the regions, and their estimates are averaged at the end to obtain the final estimate.

A. Initialization

Different initializations are compared in the context of different inverse problems, interpolation, zooming and deblurring. The reported experiments are performed on some typical image regions, Lena's hat with sharp contours and Barbara's cloth rich in texture, as illustrated in Figure 7.

Interpolation. In the addressed case of interpolation, the image is degraded by $\mathbf{U}$, that is a random masking operator which randomly sets pixel values to zeros. The initialization described above is compared with a random initialization, which initializes in the E-step all the missing pixel value with zeros and starts with a random patch clustering. Figure 7-(a) and (b) compare the PSNRs obtained by the MAP-EM algorithm with those two initializations. The algorithm with the random initialization converges to a PSNR close to, about 0.4 dB lower than, that with the proposed initialization, and the convergence takes much longer time (about 6 iterations) than the latter (about 3 iterations).

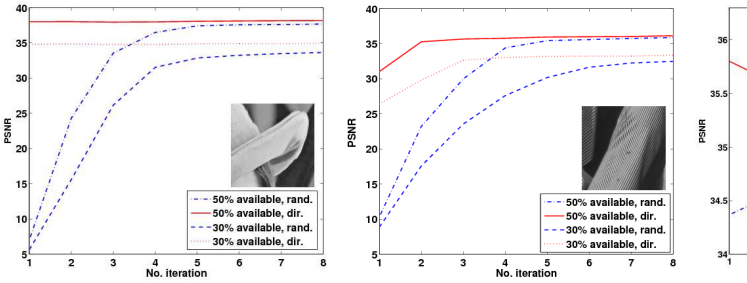
It is worth noting that on the contours of Lena's hat, with the proposed initialization the resulting PSNR is stable from the initialization, which already produces accurate estimation, since the initial directional PCA bases themselves are calculated over synthetic contour images, as described in Section III-C.

Zooming. In the context of zooming, the degradation $\mathbf{U}$ is a subsampling operator on a uniform grid, much structured than that for interpolation of randomly sampled images. The MAP-EM algorithm with the random initialization completely fails to work: It gets stuck in the initialization and does not lead to any changes on the degraded image. Instead of initializing the missing pixels with zeros, a bicubic initialization is tested, which initializes the missing pixels with bicubic interpolation. Figure 7-(c) shows that, as the MAP-EM algorithm iterates, it significantly improves the PSNR over the bicubic initialization, however, the PSNR after a slower convergence is still about 0.5 dB lower than that obtained with the proposed initialization.

Deblurring. In the deblurring setting, the degradation $\mathbf{U}$ is a convolution operator, which is very structured, and the image is further contaminated with a white Gaussian noise. Four initializations are under consideration: the initialization with

directional PCAs (K directions plus a DCT basis), which is exactly the same as that for interpolation and zooming tasks, the proposed initialization with the *position* PCA bases for deblurring as described in Section III-C2 (P positions per each of the K directions, all with the same eigenvalues as for the directional PCAs initialization), and two random initializations with the blurred image itself as the initial estimate and a random patch clustering with, respectively, $K+1$ and $(K+1)P$ clusters. As illustrated in Figure 7-(d), the algorithm with the directional PCAs initialization gets stuck in a local minimum since the second iteration, and converges to a PSNR 1.5 dB lower than that with the initialization using the position PCAs. Indeed, since the recoverability condition for deblurring, as explained in Section III-C2, is violated with just directional PCA bases, the resulting images remain still quite blurred. The random initialization with $(K+1)P$ clusters results in better results than with $K+1$ clusters, which is 0.7 dB worse than the proposed initialization with position PCAs.

These experiments confirm the importance of the initialization in the MAP-EM algorithm to solve inverse problems. The sparse modeling dual interpretation of GMM/MAP-EM helps to deduce effective initializations for different inverse problems, which are further confirmed by the Wiener filter interpretation described in Section III-C3. While for interpolation of random masking operators, trivial initializations slowly converge to a solution moderately worse than that obtained with the proposed initialization, for more structured degradation operators such as uniform subsampling and convolution, simple initializations either fail to work or lead to worse results than with the proposed initialization. Note that with the proposed initialization, the first iteration leads already to good performance. The adaptation of the PCAs to the image under consideration as the algorithm iterates further improves the results.

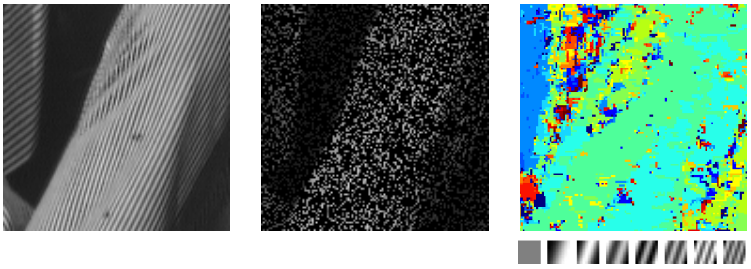


(a) **(b)**

Fig. 7. PSNR comparison of the MAP-EM algorithm with different initializations on different inverse problems. The horizontal axis corresponds to the number of iterations. (a) and (b). Interpolation with 50% and 30% available data at random position, on Lena's hat and Barbara's cloth. The initializations under consideration are the random initialization and the initialization with directional PCA bases. (c) Zooming, on Lena's hat. The initializations under consideration are bicubic initialization and the initialization with directional PCA bases. (Random initialization completely fails to work.) (d) Deblurring, on Lena's hat. The initializations under consideration are the initialization with directional PCAs (K directions plus a DCT basis), the initialization with the *position* PCA bases (P positions per each of the K directions), and two random initializations with the blurred image itself as the initial estimate and a random patch clustering with, respectively, $K+1$ (rand. 1) and $(K+1)P$ (rand. 2) clusters. See text for more details.

B. Evolution of Representations

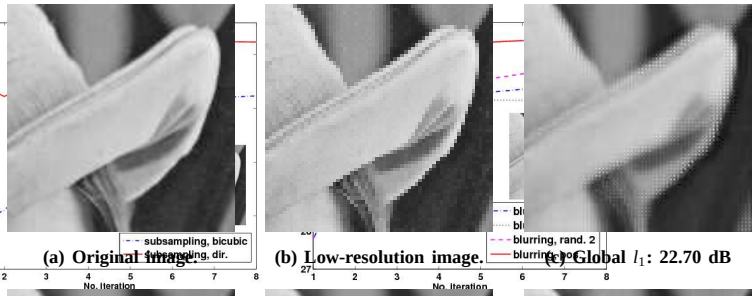
Figure 8 illustrates, in an interpolation context on Barbara's cloth, which is rich in texture, the evolution of the patch clustering as well as that of a typical PCA bases as the MAP-EM algorithm iterates. The clustering gets cleaned up as the algorithm iterates. (See figures 3-(d) and (e) for another example.) Some high-frequency atoms are promoted to better capture the oscillatory patterns, resulting in a significant PSNR improvement of more than 3 dB. On contour images such as Lena's hat illustrated in Figure 7, on the contrary, although the patch clustering is cleaned up as the algorithm iterates, the resulting local PSNR evolves little after the initialization, which already produces accurate estimation, since the directional PCA bases themselves are calculated over synthetic contour images, as described in Section III-C. The eigenvalues have always fast decay as the iteration goes on, visually similar to the plot in Figure 2-(d). The resulting PSNRs typically converge in 3 to 5 iterations.



(a) **(b)** **(c)**

Fig. 8. Evolution of the representations. (a) The original image cropped from Barbara. (b) The image masked with 30% available data. (c) and (d) are color images. (c) Bottom: The first few atoms of an initial PCA basis corresponding to the texture on the right of the image. Top: The resulting patch clustering after the 1st iteration. Different colors represent different clusters. (d) Bottom: The first few atoms of the PCA basis updated after the 1st iteration. Top: The resulting patch clustering after the 2nd iteration. (e) The interpolation estimate after the 2nd iteration (32.30 dB).

C. Estimation Methods



(a) Original image. **(b) Low-resolution image.** **(c) Global l_1: 22.70 dB**

(e) Block l_1: 26.35 dB **(f) Block OMP: 29.27 dB** **(g) Block weighted l_1: 35.94 dB**

Fig. 9. Comparison of different estimation methods on super-resolution zooming. (a) The original image cropped from Lena. (b) The low-resolution image, shown at the same scale by pixel duplication. From (c) to (h) are the super-resolution results obtained with different estimation methods. As the modeling methods get closer and closer to the proposed approach, which can be interpreted as a weighted sparse coding, results get closer and closer to the best one produced by the proposed approach, obtained at a significantly lower computational cost. See text for more details.

From the sparse coding point of view, the gain of introducing structure in sparse inverse problem estimation as described in Section III is now shown through some experiments. An overcomplete dictionary $\mathbf{D}$ composed of a family of PCA bases

$\{\mathbf{B}_k\}_{1\le k\le K}$, illustrated in Figure 1-(b), is learned as described in Section II, and is then fed to the following estimation schemes. (i) **Global l_1 and OMP**: the ensemble of $\mathbf{D}$ is used as an overcomplete dictionary, and the zooming estimation is calculated with the sparse estimate (14) through, respectively, an l_1 minimization or an orthogonal matching pursuit (OMP). (ii) **Block l_1 and OMP**: the sparse estimate is calculated in each PCA basis $\mathbf{B}_k$ through, respectively an l_1 minimization and an OMP, and the best estimate is selected with a model selection procedure similar to (7), thereby reducing the degree of freedom in the estimation with respect to the global l_1 and OMP. [80]. (iii) **Block weighted** l_1: on top of the block l_1, weights are included for each coefficient amplitude in the regularizer,

$$\tilde{\mathbf{a}}_i^k = \arg\min_{\mathbf{a}} \left(\|\mathbf{U}_i\mathbf{B}_k\mathbf{a} - \mathbf{y}_i\|^2 + \sigma^2 \sum_{m=1}^{N} \frac{|\mathbf{a}[m]|}{\tau_m^k} \right), \qquad (19)$$

with the weights $\tau_m^k = (\lambda_m^k)^{1/2}$, where λ_m^k are the eigenvalues of the k-th PCA basis. The weighting scheme penalizes the atoms that are less likely to be important, following the spirit of the weighted l_2 deduced from the MAP estimate. (iv) **Block weighted** l_2: the proposed PLE. Comparing with (19), the difference is that the weighted l_2 (18) takes the place of the weighted l_1, thereby transforming the problem into a stable and computationally efficient piecewise linear estimation.

The comparison on a typical region of Lena in the 2×2 image zooming context is shown in Figure 9. The global l_1 and OMP produce some clear artifacts along the contours, which degrade the PSNRs. The block l_1 or OMP considerably improves the results (especially for l_1). Comparing with the block l_1 or OMP, a very significant improvement is achieved by adding the collaborative weights on top of the block l_1. The proposed PLE with the block weighted l_2, computed with linear filtering, further improves the estimation accuracy over the block weighted l_1, with a much lower computational cost.

In the following sections, PLE will be applied to a number of inverse problems, including image interpolation, zooming and deblurring. The experiments are performed on some standard gray-level and color images.[3]

V. Interpolation of Random Sampled Images

In the addressed case of interpolation, the original image $\mathbf{f}$ is masked with a random mask, $\mathbf{y} = \mathbf{U}\mathbf{f}$, where $\mathbf{U}$ is a diagonal matrix whose diagonal entries are randomly either 1 or 0, keeping or killing the corresponding pixels. Note that this can be considered as a particular case of compressed sensing, or when collectively considering all the image patches, as matrix completion (and as here demonstrated, in contrast with the recent literature on the subject, a single subspace is not sufficient, see also [84]).

The experiments are performed on the gray-level images Lena, Barbara, House, and Boat, and the color images Castle, Mushroom, Train and Horses. Uniform random masks that retain 80%, 50%, 30% and 20% of the pixels are used. The masked images are then inpainted with the algorithms under consideration.

For gray-level images, the image patch size is $\sqrt{N}\times\sqrt{N} = 8\times 8$ when the available data is 80%, 50%, and 30%. Larger patches of size 12×12 are used when images are heavily masked with only 20% pixels available. For color images, patches of size $\sqrt{N}\times\sqrt{N}\times 3$ throughout the RGB color channels are used to exploit the redundancy among the channels [51]. To simplify the initialization in color image processing, the E-step in the first iteration is calculated with "gray-level" patches of size $\sqrt{N}\times\sqrt{N}$ on each channel, but with a unified model selection across the channels: The same model selection is performed throughout the channels by minimizing the sum of the model selection energy (7) over all the channels; the signal estimation is calculated in each channel separately. The M-step then estimates the Gaussian models with the "color" patches of size $\sqrt{N}\times\sqrt{N}\times 3$ based on the model selection and the signal estimate previously obtained in the E-step. Starting from the second iteration, both the E- and M-steps are calculated with "color" patches, treating the $\sqrt{N}\times\sqrt{N}\times 3$ patches as vectors of size $3N$. $\sqrt{N}$ is set to 6 for color images, as in the previous works [51], [84]. The MAP-EM algorithm runs for 5 iterations. The noise standard deviation σ is set to 3, which corresponds to the typical noise level in these images. The small constant ε in the covariance regularization is set to 30 in all the experiments.

The PLE interpolation is compared with a number of recent methods, including "MCA" (morphological component analysis) [25], "ASR" (adaptive sparse reconstructions) [37] , "ECM" (expectation conditional maximization) [29] , "KR" (kernel regression) [70], "FOE" (fields of experts) [66], "BP" (beta process) [84], "K-SVD" [51], and "NL" [45]. MCA and ECM compute the sparse inverse problem estimate in a dictionary that combines a curvelet frame [11], a wavelet frame [53] and a local DCT basis. ASR calculates the sparse estimate with a local DCT. BP infers a nonparametric Bayesian model from the image under test (noise level is automatically estimated). Using a natural image training set, FOE and K-SVD learn respectively a Markov random field model and an overcomplete dictionary that gives sparse representation for the images. Following the self-similarity image prior, NL iterates between a projection step based on the observation, and a non-local transform and thresholding step. The results of MCA, ECM, KR, FOE, and NL are generated by the original authors' softwares, with the parameters manually optimized over all the images, and those of ASR are calculated with our own implementation. The PSNRs of BP and K-SVD are cited from the corresponding papers. NL, BP, and BK-SVD currently generate the best interpolation results in the literature.

Table I-left gives the interpolation results on gray-level images. Except at relatively high available data ratio (80% and 50%) where NL gives the results comparable to PLE, PLE considerably outperforms the other methods in all the cases, with an average PSNR improvement of about 0.5 dB over the second best algorithm NL and about 2 dB over the algorithms that follow (BP, FOE and MCA). With 20% available data on Barbara, which is rich in textures, it gains as much as about 3 dB over MCA, 4 dB over ECM, 5.5 dB over NL, and 6 dB over all the other methods. Let us remark that when the missing data ratio is high, MCA generates quite good results, as it benefits from the curvelet atoms that have large support relatively to the local patches used by the other methods.

Figure 10 compares the results of different algorithms. All the methods lead to good interpolation results on the smooth regions. MCA is good at capturing contour structures. However,

[3]Gray-level: Lena, Barbara, Peppers, Mandril, House, Cameraman, Boats, and Straws; Color images: Castle, Mushroom, Kangaroo, Train, Horses, Kodak05, Kodak20, Girl, and Flower.

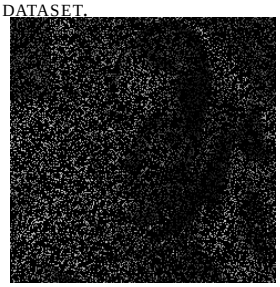
when the curvelet atoms are not correctly selected, MCA produces noticeable elongated curvelet-like artifacts that degrade the visual quality and offset its gain in PSRN (see for example the face of Barbara). MCA restores better textures than BP, ASR, FOE, KR, and NL. PLE leads to accurate restoration on both the directional structures and the textures, producing the best visual quality with the highest PSNRs. An additional PLE interpolation examples is shown in Figure 8.

Data ratio		MCA	ASR	ECM	KR	FOE*	NL	BP	PLE
Lena	80%	40.60	42.18	39.51	41.68	42.17	43.30	41.27	**42.38**
	50%	35.63	36.16	34.43	36.77	36.66	**37.82**	36.94	37.78
	30%	32.33	32.48	31.11	33.55	33.22	34.13	33.31	**34.37**
	20%	30.30	30.37	28.93	31.21	31.06	31.62	31.00	**32.22**
Barbara	80%	41.50	39.63	39.10	37.81	38.27	43.50	40.76	**43.85**
	50%	34.29	30.42	32.54	27.98	29.47	36.40	33.17	**37.03**
	30%	29.98	25.72	28.46	24.00	25.36	29.65	27.52	**32.73**
	20%	27.47	24.66	26.45	23.34	23.93	25.40	24.80	**30.54**
House	80%	42.91	43.79	40.61	42.57	44.70	**45.27**	43.03	44.77
	50%	37.02	36.06	35.16	36.82	37.99	**39.30**	38.02	38.97
	30%	33.41	31.86	31.46	33.62	33.86	**35.92**	33.14	34.88
	20%	30.67	29.91	28.97	31.19	31.28	32.87	30.12	**33.05**
Boat	80%	38.61	39.52	37.45	37.91	38.33	**40.52**	39.50	40.49
	50%	32.77	32.84	31.84	32.70	33.22	**34.57**	33.78	34.36
	30%	29.57	29.55	28.46	29.28	29.80	30.51	30.00	**30.77**
	20%	27.73	27.34	26.39	27.05	27.86	28.32	27.81	**28.66**
Average	80%	40.90	41.28	39.16	39.99	40.86	**43.16**	41.14	4
	50%	34.93	33.87	33.49	33.56	34.33	37.02	35.47	3
	30%	31.32	29.90	29.87	30.11	30.56	32.55	30.99	3
	20%	29.04	28.07	27.68	28.19	28.53	29.55	28.43	3

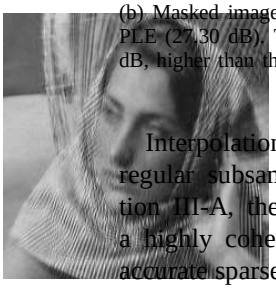
TABLE I

PSNR COMPARISON ON GRAY-LEVEL (LEFT) AND COLOR (RIGHT) IMAGE INTERPOLATION. FOR EACH IMAGE, UNIFORM RANDOM MASKS WITH FOUR AVAILABLE DATA RATIOS ARE TESTED. THE ALGORITHMS UNDER CONSIDERATION ARE MCA [25], ASR [37] , ECM [29] , KR [70], FOE [66], BP [84], NL [45], AND THE PROPOSED PLE FRAMEWORK. THE BOTTOM BOX SHOWS THE AVERAGE PSNRS GIVEN BY EACH METHOD OVER ALL THE IMAGES AT EACH AVAILABLE DATA RATIO. THE HIGHEST PSNR IN EACH ROW IS IN BOLDFACE. THE ALGORITHMS WITH * USE A TRAINING DATASET.

(a) Original



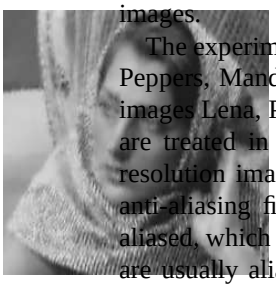
(b) Masked



(c) MCA (24.18 dB)

(e) FOE (21.92 dB)

(f) NL (23.31 dB)



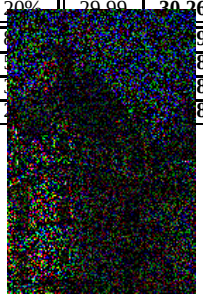
(g) BP (25.54 dB)

Fig. 10. Gray-level image interpolation. (a) Original image cropped from Barbara. (b) Masked image with 20% available data (6.81 dB). From (c) to (g): Image inpainted by different algorithms. Note the overall superior visual quality obtained with the proposed approach. The PSNRs are calculated on the cropped images.

Table I-right compares the PSNRs of the PLE color image interpolation results with those of BP (the only one in the literature that reports the comprehensive comparison in our knowledge). Again, PLE generates higher PSNRs in all the cases. While the gain is especially large, at about 6 dB, when the available data ratio is high (at 80%), for the other masking rates, it is mostly between 0.5 and 1 dB. Both methods use only the image under test to learn the dictionaries.

Figure 11 illustrates the PLE interpolation result on Castle with 20% available data. Calculated with a much reduced computational complexity, the resulting 30.07 dB PSNR surpasses the highest PSNR, 29.65 dB, reported in the literature, produced by K-SVD [51], that uses a dictionary learned from a natural image training set, followed by 29.12 dB given by BP (BP has been recently improved at the cost of coherence in the code, unpublished results). As shown in the zoomed region, PLE accurately restores the details of the castle from the heavily masked image. Let us remark that interpolation with random masks on color images is in general more favorable than on gray-level images, thanks to the information redundancy among the color channels. A further comparison with a multiscale extension of the K-SVD algorithm [52] shows that for restoring House from 25% available data (the only result of this application reported therein), the multiscale K-SVD leads to 33.97 and 31.75 dB at respectively two and one scales, in contrast to the 34.05 dB obtained by the proposed PLE without any parameter tuning.

(a) Original



(b) Masked

(c) PLE

Fig. 11. Color image interpolation. (a) Original image cropped from Castle. (b) Masked image with 20% available data (5.44 dB). (c) Image inpainted by PLE (27.30 dB). The PSNR on the overall image obtained with PLE is 30.07 dB, higher than the best result reported so far in the literature 29.65 dB [51].

VI. INTERPOLATION ZOOMING

Interpolation zooming is a special case of interpolation with regular subsampling on uniform grids. As explained in Section III-A, the regular subsampling operator **U** may result in a highly coherent transformed dictionary **UD**. Calculating an accurate sparse estimation for interpolation zooming is therefore more difficult than that for interpolation of random sampled images.

The experiments are performed on the gray-level images Lena, Peppers, Mandril, Cameraman, Boat, and Straws, and the color images Lena, Peppers, Kodak05 and Kokad20. The color images are treated in the same way as for interpolation. These high-resolution images are down-sampled by a factor 2×2 without anti-aliasing filtering. The resulting low-resolution images are aliased, which corresponds to the reality of television images that are usually aliased, since this improves their visual perception. The low-resolution images are then zoomed by the algorithms under consideration. When the anti-aliasing blurring operator is included before subsampling, zooming can be casted as a deconvolution problem and will be addressed in Section VII.

The PLE interpolation zooming is compared with linear interpolators [8], [42], [74], [60] as well as recent super-resolution algorithms “NEDI” (new edge directed interpolation) [46], “DFDF” (directional filtering and data fusion) [81], “KR” (kernel regression) [70], “ECM” (expectation conditional

maximization) [29], “Contourlet” [59], “ASR” (adaptive sparse reconstructions) [37], “FOE” (fields of experts) [66], “SR” (sparse representation) [76], “NL” [45], “SAI” (soft-decision adaptive Interpolation) [82] and “SME” (sparse mixing estimators) [55]. KR, ECM, ASR, FOE and NL are generic interpolation algorithms that have been described in Section V. NEDI, DFDF and SAI are adaptive directional interpolation methods that take advantage of the image directional regularity. Contourlet is a sparse inverse problem estimator as described in Section III-A, computed in a contourlet frame. SR is also a sparse inverse estimator that learns the dictionaries from a training image set. SME is a recent zooming algorithm that exploits directional structured sparsity in wavelet representations. Among the previously published algorithms, SAI and SME currently provide the best PSNR for spatial image interpolation zooming [55], [82]. The results of ASR are generated with our own implementation, and those of all the other algorithms are produced by the original authors’ softwares, with the parameters manually optimized. As the anti-aliasing operator is not included in the interpolation zooming model, to obtain correct results with SR, the anti-aliasing filter used in the original authors’ SR software is deactivated in both dictionary training (with the authors’ original training dataset of 92 images) and super-resolution estimation. PLE is configured in the same way as for interpolation as described in Section V, with patch size 8×8 for gray-level images, and $6 \times 6 \times 3$ for color images.

Table II gives the PSNRs generated by all algorithms on the gray-level and the color images. Bicubic interpolation provides nearly the best results among all tested linear interpolators, including cubic splines [74], MOMS [8] and others [60], due to the aliasing produced by the down-sampling. PLE gives moderately higher PSNRs than SME and SAI for all the images, with one exception where the SAI produces slightly higher PSNR. Their gain in PSNR is significantly larger than with all the other algorithms.

Figure 12 compares an interpolated image obtained by the baseline bicubic interpolation and the algorithms that generate the highest PSNRs, SAI and PLE. The local PSNRs on the cropped images produced by all the methods under consideration are reported as well. Bicubic interpolation produces some blur and jaggy artifacts in the zoomed images. These artifacts are reduced to some extent by the NEDI, DFDF, KR, FOE and NL algorithms, but the image quality is still lower than with PLE, SAI and SME algorithms, as also reflected in the PSNRs. SR yields an image that looks sharp. However, due to the coherence of the transformed dictionary, as explained in Section III-A, when the approximating atoms are not correctly selected, it produces artifact patterns along the contours, which degrade its PSNR. The PLE algorithm restores slightly better than SAI and SME on regular geometrical structures, as can be observed on the upper and lower propellers, as well as on the fine lines on the side of the plane indicated by the arrows.

VII. Deblurring

An image $\mathbf{f}$ is blurred and contaminated by additive noise, $\mathbf{y} = \mathbf{U}\mathbf{f} + \mathbf{w}$, where $\mathbf{U}$ is a convolution operator and $\mathbf{w}$ is the noise. Image deblurring aims at estimating $\mathbf{f}$ from the blurred and noisy observation $\mathbf{y}$.

	Bicubic	NEDI	DFDF	KR	ECM	Contourlet	ASR	FOE
Lena	33.93	33.77	33.91	33.94	24.31	33.92	33.19	34.0
Peppers	32.83	33.00	33.18	33.15	23.60	33.10	32.33	31.9
Mandril	22.92	23.16	22.83	22.93	20.34	22.53	22.66	22.9
Cameraman	25.37	25.42	25.67	25.51	19.50	25.35	25.33	25.5
Boat	29.24	29.19	29.32	29.18	22.20	29.25	28.96	29.3
Straws	20.53	20.54	20.70	20.76	17.09	20.52	20.54	20.4
Ave. gain	0	0.04	0.13	0.11	-6.30	-0.02	-0.30	-0.0

	Bicubic	NEDI	DFDF	KR	FOE*	SR*	SA
Lena	32.41	32.47	32.46	32.55	32.55	26.42	32.
Peppers	30.95	31.06	31.24	31.26	31.05	26.43	31.
Kodak05	25.82	25.93	26.03	26.09	26.01	20.76	**26.**
Kodak20	30.65	31.06	31.08	30.97	30.84	25.92	31.
Ave. gain	0	0.17	0.25	0.27	0.16	-5.07	0.7

TABLE II
PSNR COMPARISON ON GRAY-LEVEL (TOP) AND COLOR (BOTTOM) IMAGE INTERPOLATION ZOOMING. THE ALGORITHMS UNDER CONSIDERATION ARE BICUBIC INTERPOLATION, NEDI [46], DFDF [81], KR [70], ECM [29], CONTOURLET [59], ASR [37], FOE [66], SR [76], NL [45] SAI [82] , SME [55] AND THE PROPOSED PLE FRAMEWORK. THE BOTTOM ROW SHOWS THE AVERAGE GAIN OF EACH METHOD RELATIVE TO THE BICUBIC INTERPOLATION. THE HIGHEST PSNR IN EACH ROW IS IN BOLDFACE. THE WITH * USE A TRAI

(a) HR **(b) LR** **(c) Bibubic**

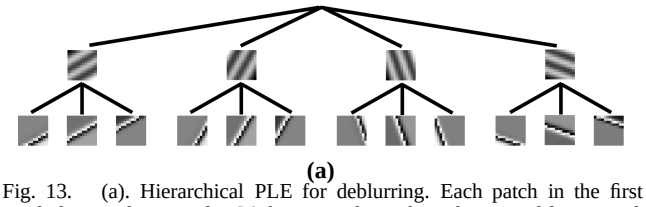
Fig. 12. Color image zooming. (a) Crop from the high-resolution image Kodak20. (b) Low-resolution image. From (c) to (e), images zoomed by bicubic interpolation (28.48 dB), SAI (30.32 dB) [82], and proposed PLE framework (30.64 dB). PSNRs obtained by the other methods under consideration: NEDI (29.68 dB) [46], DFDF (29.41 dB) [81], KR (29.49 dB) [70], FOE (28.73 dB) [66], SR (23.85 dB) [76], and SME (29.90 dB) [55]. Attention should be focused on the places indicated by the arrows.

A. Hierarchical PLE

As explained in Section III-C2, the recoverability condition of sparse super-resolution estimates for deblurring requires a dictionary comprising atoms with spread Fourier spectrum and thus localized in space, such as the position PCA basis illustrated in Figure 4. To reduce the computational complexity, model selection with a hierarchy of directional PCA bases and position PCA bases is proposed, in the same spirit of [79]. Figure 13-(a) illustrates the hierarchical PLE with a cascade of the two layers of model selections. The first layer selects the direction, and given the direction, the second layer further specifies the position.

In the first layer, the model selection procedure is identical to that in image interpolation and zooming, i.e., it is calculated with the Gaussian models corresponding to the directional PCA bases $\{\mathbf{B}_k\}_{1 \le k \le K}$, Figure 2-(c). In this layer, a directional PCA $\mathbf{B}_k$ of orientation θ is selected for each patch. Given the directional basis $\mathbf{B}_k$ selected in the first layer, the second layer recalculates the model selection (7), this time with a family of position PCA bases $\{\mathbf{B}_{k,p}\}_{1 \le p \le P}$ corresponding to the same direction θ as the directional basis $\mathbf{B}_k$ selected in the first layer, with atoms in each basis $\mathbf{B}_{k,p}$ localized at one position, and the P

bases translating in space and covering the whole patch. The image patch estimation (8) is obtained in the second layer. This hierarchical calculation reduces the computational complexity from $\mathcal{O}(KP)$ to $\mathcal{O}(K+P)$.



(a)

Fig. 13. (a). Hierarchical PLE for deblurring. Each patch in the first layer symbolizes a directional PCA basis. Each patch in the second layer symbolizes a position PCA basis. (b) To circumvent boundary issues, deblurring a patch whose support is Ω can be casted as inverting an operator compounded by a masking and a convolution defined on a larger support $\bar{\Omega}$. See text for details.

For deblurring, boundary issues on the patches need to be addressed. Since the convolution operator is non-diagonal, the deconvolution of each pixel $\mathbf{y}(x)$ in the blurred image $\mathbf{y}$ involves the pixels in a neighborhood around x whose size depends on the blurring kernel. As the patch based methods deal with the local patches, for a given patch, the information outside of it is missing. Therefore, it is impossible to obtain accurate deconvolution estimation on the boundaries of the patches. To circumvent this boundary problem, a larger patch is considered and the deconvolution is casted as a deconvolution plus an interpolation problem. Let us retake the notations $\mathbf{f}_i$, $\mathbf{y}_i$ and $\mathbf{w}_i$ to denote respectively the patches of size $\sqrt{N}\times\sqrt{N}$ in the original image $\mathbf{f}$, the degraded image $\mathbf{y}$, and the noise $\mathbf{w}$. Let Ω be their support. Let $\bar{\mathbf{f}}_i$, $\bar{\mathbf{y}}_i$ and $\bar{\mathbf{w}}_i$ be the corresponding larger patches of size $(\sqrt{N}+2r)\times(\sqrt{N}+2r)$, whose support $\bar{\Omega}$ is centered at the same position as Ω and with an extended boundary $\bar{\Omega}\backslash\Omega$ of width r (the width of the blurring kernel, see below), as illustrated in Figure 13-(b). Let $\bar{\mathbf{U}}$ be an extension of the convolution operator $\mathbf{U}$ on $\bar{\Omega}$ such that $\bar{\mathbf{U}}\bar{\mathbf{f}}_i(x)=\mathbf{U}\mathbf{f}_i(x)$ if $x\in\Omega$, and 0 if $x\in\bar{\Omega}\backslash\Omega$. Let $\mathbf{M}$ be a masking operator defined on $\bar{\Omega}$ which keeps all the pixels in the central part Ω and kills the rest, i.e., $\mathbf{M}\bar{\mathbf{f}}_i(x)=\bar{\mathbf{f}}_i(x)$ if $x\in\Omega$, and 0 if $x\in\bar{\Omega}\backslash\Omega$. If the width r of the boundary $\bar{\Omega}\backslash\Omega$ is larger than the radius of the blurring kernel, one can show that the blurring operation can be rewritten locally as an extended convolution on the larger support followed by a masking, $\mathbf{M}\bar{\mathbf{y}}_i=\mathbf{M}\bar{\mathbf{U}}\bar{\mathbf{f}}_i+\mathbf{M}\bar{\mathbf{w}}_i$. Estimating $\mathbf{f}_i$ from $\mathbf{y}_i$ can thus be calculated by estimating $\bar{\mathbf{f}}_i$ from $\mathbf{M}\bar{\mathbf{y}}_i$, following exactly the same algorithm, now treating the compounded $\mathbf{MU}$ as the degradation operator to be inverted. The boundary pixels in the estimate $\tilde{\mathbf{f}}_i(x)$, $x\in\bar{\Omega}\backslash\Omega$, can be interpreted as an extrapolation from $\mathbf{y}_i$, therefore less reliable. The deblurring estimate $\tilde{\mathbf{f}}_i$ is obtained by discarding these boundary pixels from $\tilde{\mathbf{f}}_i$ (which are outside of Ω anyway).

Local patch based deconvolution algorithms become less accurate if the blurring kernel support is large relative to the patch size. In the deconvolution experiments reported below, Ω and $\bar{\Omega}$ are respectively set to 8×8 and 12×12. In the initialization the number of directions is set to $K=18$, the same as in the image interpolation and zooming experiments, and $P=16$ positions is set for each direction. The blurring kernels are restricted to a 5×5 support.

B. Deblurring Experiments

The deblurring experiments are performed on the gray-level images Lena, Barbara, Boat, House, and Cameraman, with different amounts of blur and noise. The PLE deblurring is compared with a number of deconvolution algorithms: "ForWaRD" (Fourier-wavelet regularized deconvolution) [61], "TVB" (total variation based) [7], "TwIST" (two-step iterative shrinkage/thresholding) [6], "SP" (sparse prior) [44], "SA-DCT" (shape adaptive DCT) [30], "BM3D" (3D transform-domain collaborative filtering) [18], and "DSD" (direction sparse deconvolution) [48]. ForWaRD, SA-DCT and BM3D first calculate the deconvolution with a regularized Wiener filter in Fourier, and then denoise the Wiener estimate with, respectively, a thresholding estimator in wavelet and SA-DCT representations, and with the non-local 3D collaborative filtering [17]. TVB and TwIST deconvolutions regularize the estimate with the image total variation prior. SP assumes a sparse prior on the image gradient. DSD is a recently developed sparse inverse problem estimator, described in Section III-A. In the previous published works, BM3D and SA-DCT are among the deblurring methods that produce the highest PSNRs, followed by SP. The results of all the methods under comparison are generated by the authors' original softwares, with the parameters manually optimized. The proposed algorithm runs for 5 iterations.

Table III gives the ISNRs (improvement in PSNR relative to the input image) of the different algorithms for restoring images blurred with Gaussian kernels of standard deviation $\sigma_b=1$ and 2 (truncated to a 5×5 support), and 5×5 uniform box kernel, all then contaminated by a white Gaussian noise of standard deviation $\sigma_n=5$. [4] BM3D produces the highest ISNRs, followed closely in the case of Gaussian blurring kernels by SA-DCT and PLE, whose ISNRs are comparable and are moderately higher than with SP on average. As the more aggressive uniform box kernel is tested, the local patch-based PLE is outperformed by most methods under comparison, which calculates the deblurring on the whole image instead of patch by patch. Since the convolution operator is non-diagonal, it leads to a border effect in the deblurred image or image patches. Such border effect may dominate in local patches as the kernel size increases, degrading the performance of patch-based deblurring method. The same is observed with other patch-based deblurring algorithms as well [48]. Let us remark that BM3D, SA-DCT and ForWaRD include an empirical Wiener filtering post-processing that, as reported in the table, boosts the ISNR on average from 0.3 to 2 dB, leading to state-of-the-art results for the case of BM3D.

Figure 14 shows a deblurring example. All the algorithms under consideration reduce the amount of blur and attenuate the noise. BM3D generates the highest ISNR, followed by SA-DCT, PLE and SP, all producing similar visual quality, which are moderately better than the other methods. DSD accurately restores sharp image structures when the atoms are correctly selected, however, some artifacts due to the incorrect atom selection offset its gain in ISNR. As a core component in BM3D and SA-DCT, the empirical Wiener filtering efficiently removes some artifacts and significantly improves the visual quality and the ISNR. More examples of PLE deblurring will be shown in the next section.

[4] When calculating the ISNRs, the image borders of width 6 pixels are removed in order to eliminate the boundary effects.

Kernel size and input PSNR			ForWaRD	TVB	TwIST	SA-DCT	BM3D	SP	DSD*	PLE
Lena	$\sigma_b = 1$	30.62	2.20/2.16	3.03	2.87	3.56/2.58	**4.03**/3.45	3.31	2.56	*3.77*
	$\sigma_b = 2$	28.84	2.50/2.51	3.15	3.13	3.46/3.00	**3.91**/3.20	3.40	2.47	3.52
	Box 5	28.19	2.79/2.69	3.25	3.22	3.86/3.22	**4.11**/3.26	3.48	NA	3.67
House	$\sigma_b = 1$	30.04	2.60/2.18	3.12	3.23	4.14/3.07	4.29/3.60	3.52	2.27	**4.30**
	$\sigma_b = 2$	28.02	3.16/2.79	3.24	3.82	4.21/3.64	**4.73**/4.11	3.92	2.97	3.90
	Box 5	27.26	3.86/3.29	4.20	4.60	5.22/4.29	**5.46**/4.58	4.57	NA	4.26
Boat	$\sigma_b = 1$	28.29	2.01/1.68	2.45	2.44	2.93/2.21	**3.23**/2.46	2.70	1.93	*2.72*
	$\sigma_b = 2$	26.21	2.44/2.12	2.67	2.59	3.71/*2.63*	**3.33**/2.44	2.60	2.02	2.48
	Box 5	25.45	3.04/2.62	3.18	3.21	3.91/3.24	**4.07**/3.16	3.29	NA	2.62
Average	$\sigma_b = 1$	29.65	2.27/2.01	2.87	2.84	3.54/2.62	**3.85**/3.33	3.17	2.25	3.60
	$\sigma_b = 2$	27.69	2.70/2.47	3.02	3.18	3.79/3.09	**3.99**/3.25	3.30	2.48	*3.31*
	Box 5	27.69	3.23/2.87	3.54	3.68	4.33/3.58	**4.54**/2.48	*3.81.*	NA	3.32

TABLE III
ISNR (IMPROVEMENT IN PSNR WITH RESPECT TO INPUT IMAGE) COMPARISON ON IMAGE DEBLURRING. IMAGES ARE BLURRED BY A GAUSSIAN KERNEL OF STANDARD DEVIATION $\sigma_b = 1$ AND 2, AND A 5×5 UNIFORM BOX KERNEL, AND ARE THEN CONTAMINATED BY WHITE GAUSSIAN NOISE OF STANDARD DEVIATION $\sigma_n = 5$. FROM LEFT TO RIGHT: FORWARD (WITH/WITHOUT EMPIRICAL WIENER POST-PROCESSING) [61], TVB [7], TWIST [6], SA-DCT (WITH/WITHOUT EMPIRICAL WIENER POST-PROCESSING) [30], BM3D (WITH/WITHOUT EMPIRICAL WIENER POST-PROCESSING) [18], SP [44], DSD [48], AND THE PROPOSED PLE FRAMEWORK. THE BOTTOM BOX SHOWS THE AVERAGE ISNRS GIVEN BY EACH METHOD OVER ALL THE IMAGES WITH DIFFERENT AMOUNTS OF BLUR. THE HIGHEST ISNR IN EACH ROW IS IN BOLDFACE, WHILE THE HIGHEST WITHOUT POST-PROCESSING IS IN ITALIC. THE ALGORITHMS WITH * USE A TRAINING DATASET.

(a) Original **(b) Blurred and noisy** **(c) BM3D** **(d) PLE**

Fig. 14. Gray-level image deblurring. (a) Crop from Lena. (b) Image blurred by a Gaussian kernel of standard deviation $\sigma_b = 1$ and contaminated by white Gaussian noise of standard deviation $\sigma_n = 5$ (PSNR=27.10). (c) and (d). Images deblurred by BM3D with empirical Wiener post-processing (ISNR 3.40 dB dB) [18], and the proposed PLE framework (ISNR 2.94 dB).

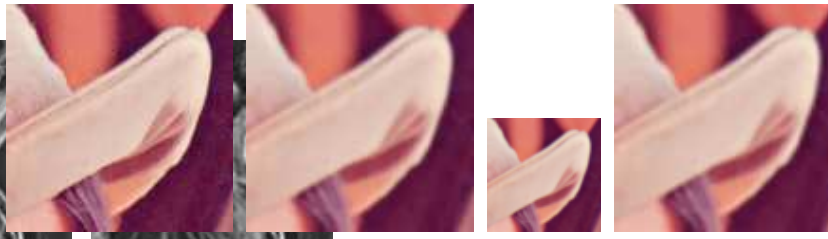
C. Zooming deblurring

When an anti-aliasing filtering is taken into account, image zooming-out can be formulated as $\mathbf{y} = \mathbf{SUf}$, where $\mathbf{f}$ is the high-resolution image, $\mathbf{U}$ and $\mathbf{S}$ are respectively an anti-aliasing convolution and a subsampling operator, and $\mathbf{y}$ is the resulting low-resolution image. Image zooming aims at estimating $\mathbf{f}$ from $\mathbf{y}$, which amounts to inverting the combination of the two operators $\mathbf{S}$ and $\mathbf{U}$.

Image zooming can be calculated differently under different amounts of blur introduced by $\mathbf{U}$. Let us distinguish between three cases: (i) If the anti-aliasing filtering $\mathbf{U}$ removes enough high-frequencies from $\mathbf{f}$ so that $\mathbf{y} = \mathbf{SUf}$ is free of aliasing, then the subsampling operator $\mathbf{S}$ can be perfectly inverted with a linear interpolation denoted as $\mathbf{I}$, i.e., $\mathbf{IS} = Id$ [53]. In this case, zooming can be calculated as a deconvolution problem on $\mathbf{Iy} = \mathbf{Uf}$, where one seeks to invert the convolution operator $\mathbf{U}$. In reality, however, camera and television images contain, always a certain amount of aliasing, since it improves the visual perception, i.e., the anti-aliasing filtering $\mathbf{U}$ does not eliminate all the high-frequencies from $\mathbf{f}$. (ii) When $\mathbf{U}$ removes a small amount of high-frequencies, which is often the case in reality, zooming can be casted as an interpolation problem [46], [55], [59], [70], [81], [82], where one seeks to invert only $\mathbf{S}$, as addressed in Section VI. (iii) When $\mathbf{U}$ removes an intermediate amount of blur from $\mathbf{f}$, the optimal zooming solution is inverting $\mathbf{SU}$ together as a compounded operator, as investigated in [76].

This section introduces a possible solution for the case (iii) with the PLE framework. A linear interpolation $\mathbf{I}$ is first applied to partially invert the subsampling operator $\mathbf{S}$. Due to the aliasing, the linear interpolation does not perfectly restore $\mathbf{Uf}$, nevertheless, it remains rather accurate, i.e., the interpolated image $\mathbf{Iy} = \mathbf{ISUf}$ is close to the blurred image $\mathbf{Uf}$, as $\mathbf{Uf}$ has mainly low frequencies in the case (iii). The PLE deblurring framework is then applied to deconvolve $\mathbf{U}$ from $\mathbf{Iy}$. Inverting the operator $\mathbf{U}$ is simpler than inverting the compounded operator $\mathbf{SU}$. As the linear interpolation $\mathbf{I}$ in the first step is accurate enough in the case (iii), deconvolving $\mathbf{Iy}$ results in accurate zooming estimates.

In the experiments below, the anti-aliasing filter $\mathbf{U}$ is set as a Gaussian convolution of standard deviation $\sigma_G = 1.0$ and $\mathbf{S}$ is an $s \times s = 2 \times 2$ subsampling operator. It has been shown that a pre-filtering with a Gaussian kernel of $\sigma_G = 0.8s$ guarantees that the following $s \times s$ subsampling generates a quasi aliasing-free image [58]. For a 2×2 subsampling, the anti-aliasing filtering $\mathbf{U}$ with $\sigma_G = 1.0$ leads to an amount of aliasing and visual quality comparable to that in typical camera pictures in reality.



(a) f **(b) Uf** **(c) y = SUf** **(d) Iy**

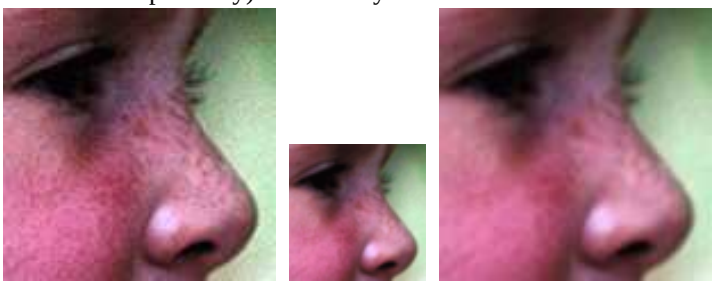
Fig. 15. Color image zooming deblurring. (a) Crop from Lena: $\mathbf{f}$. (b) Image pre-filtered with a Gaussian kernel of standard deviation $\sigma_G = 1.0$: $\mathbf{Uf}$. (c) Image subsampled from $\mathbf{Uf}$ by a factor of 2×2: $\mathbf{y} = \mathbf{SUf}$. (d) Image interpolated from $\mathbf{y}$ with a cubic spline interpolation: $\mathbf{Iy}$ (31.03 dB). (e) Image deblurred from $\mathbf{Iy}$ by the proposed PLE framework (34.27 dB). (f) Image zoomed from $\mathbf{y}$ with [76] (29.66 dB). The PSNRs are calculated on the cropped image between the original $\mathbf{f}$ and the one under evaluation.

Figure 15 illustrates an experiment on the image Lena. Figures 15-(a) to (c) show, respectively, a crop of the original image $\mathbf{f}$, the pre-filtered version $\mathbf{Uf}$, and the low-resolution image after subsampling $\mathbf{y} = \mathbf{SUf}$. As the amount of aliasing is limited in $\mathbf{y}$ thanks to the anti-aliasing filtering, a cubic spline interpolation is more accurate than lower ordered interpolations such as bicubic [74], and is therefore applied to upsample $\mathbf{y}$, the resulting image $\mathbf{Iy}$ illustrated in Figure 15-(d). A visual inspection confirms that $\mathbf{Iy}$ is very close to $\mathbf{Uf}$, the PSNR between them being as high as 50.02 dB. The PLE deblurring is then applied to calculate the final zooming estimate $\tilde{\mathbf{f}}$ by deconvolving $\mathbf{U}$ from $\mathbf{Iy}$. (As no noise is added after the anti-aliasing filter, the noise standard deviation is set to a small value $\sigma = 1$.) As illustrated in Figure 15-(e), the resulting image $\mathbf{f}$ is much sharper, without noticeable artifacts, and improves by 3.12 dB with respect to $\mathbf{Iy}$. Figure 15-(f) shows the result obtained with "SR" (sparse representation) [76]. SR implements a sparse inverse problem estimator that tries to invert the compounded operator $\mathbf{SU}$, with a dictionary learned from a natural image dataset. The experiments were performed with the authors' original software and training image set. The dictionaries were retrained with the $\mathbf{US}$ described above. It can be observed that the resulting image looks sharper and the restoration is accurate when the atoms selection is correct. However, due to the coherence of the dictionaries as explained in Section III-A, some noticeable artifacts along the edges are produced when the atoms are incorrectly selected, which also offset its gain in

PSNR.

Figure 16 shows another set of experiments on the image Girl. Again PLE efficiently reduces the blur from the interpolated image and leads to a sharp zoomed image without noticeable artifacts. SR produces similarly good visual quality as PLE, however, some slight but noticeable artifacts (near the end of the nose for example) due to the incorrect atom selection offset its gain in PSNR.

Comparing the PSNR for the color image images Lena, Girl and Flower. PLE deblurring from the cubic spline interpolation improves from 1 to 2 dB PSNR over the interpolated images (33.78, 31.82, and 39.06 dB respectively for PLE; and 31.60, 30.62, and 37.02 dB for the cubic spline). Although SR is able to restore sharp images, its gain in PSNR (30.64, 30.43, and 35.96 dB respectively) is offset by the noticeable artifacts.



(a) HR **(b) LR** **(c) Cubic spline** **(d) PLE** **(e) SR**

Fig. 16. Color image zooming deblurring. (a) Crop from Girl: **f**. (b) Image prefiltered with a Gaussian kernel of standard deviation $\sigma_G = 1.0$, and subsampled by a factor of 2×2: $\mathbf{y} = \mathbf{SUf}$. (c) Image interpolated from **y** with a cubic spline interpolation: **Iy** (29.40 dB). (d) Image deblurred from **Iy** by the proposed PLE framework (30.49 dB). (e) Image zoomed from **y** with [76] (28.93 dB).

VIII. Conclusion and Future Works

This work has shown that a piecewise linear estimation (PLE) based on Gaussian mixture models (GMM) and calculated with a MAP-EM algorithm provides general and effective solutions for inverse problems, leading to results in the same ballpark as state-of-the-art ones in various image inverse problems. A dual mathematical interpretation of the framework with structured sparse estimation is described, which shows that PLE stabilizes and improves the traditional fully non-linear sparse inverse problem approaches. This connection also suggests an effective dictionary motivated initialization for the MAP-EM algorithm. In a number of image restoration applications, including interpolation, zooming, and deblurring of narrow kernels, the same simple (its core is formulated in four equations (5), (7), (8), and (10), implementing a MAP and a ML estimations) and computationally efficient algorithm produces results in the same ballpark as the state-of-the-art, with a reduced computational complexity than other popular leading algorithms, e.g., l_1 sparse estimations. The proposed PLE has also been applied to image denoising (**U** being the identity matrix), achieving good performance (see the full presentation [78] for details and comparisons with leading algorithms such as those in [17], [50] in the noise standard deviation range of $[5, 25]$).

A theoretical study considering Gaussian models in compressed sensing is being undertaken [77], and applications of Gaussian models as those here developed have been extended to matrix completion problems [43].

Acknowledgements: *Work supported by NSF, NGA, ONR, ARO and NSSEFF. We thank Stéphanie Allassonnière for helpful discussions, in particular about MAP-EM and covariance regularization.*

Guoshen Yu received the B.Sc. degree in electronic engineering from Fudan University, Shanghai, China, in 2003, the engineering degree from Telecom ParisTech, France, in 2006, the M.Sc. degree in applied mathematics from ENS de Cachan, France, in 2006, and the Ph.D. degree in applied mathematics from Ecole Polytechnique, France, in 2009.

He was a Postdoctoral Research Associate at the Electrical and Computer Engineering Department, University of Minnesota, Twin Cities in 2009-2010. In the spring 2008 semester, he was a visiting graduate student in the Mechanical Engineering Department, Massachusetts Institute of Technology (MIT), Cambridge. Since December 2010, he has been a quantitative researcher at BlueCrest Capital Management, Geneva. His research interests include image and signal processing, and its applications in finance.

Dr. Yu received the ParisTech Doctoral Thesis Award in 2010, and the Gaspard Monge International Doctoral Grant from Ecole Polytechnique from 2006 to 2008.

Guillermo Sapiro was born in Montevideo, Uruguay, on April 3, 1966. He received his B.Sc. (summa cum laude), M.Sc., and Ph.D. from the Department of Electrical Engineering of the Technion, Israel Institute of Technology, in 1989, 1991, and 1993, respectively. After post-doctoral research at MIT, Dr. Sapiro became Member of Technical Staff at the research facilities of HP Labs in Palo Alto, California. He is currently with the Department of Electrical and Computer Engineering at the University of Minnesota, where he holds the position of Distinguished McKnight University Professor and Vincentine Hermes-Luh Chair in Electrical and Computer Engineering.

G. Sapiro was awarded the Gutwirth Scholarship for Special Excellence in Graduate Studies in 1991, the Ollendorff Fellowship for Excellence in Vision and Image Understanding Work in 1992, the Rothschild Fellowship for Post-Doctoral Studies in 1993, the Office of Naval Research Young Investigator Award in 1998, the Presidential Early Career Awards for Scientist and Engineers (PECASE) in 1998, the National Science Foundation Career Award in 1999, and the National Security Science and Engineering Faculty Fellowship in 2010.

G. Sapiro is the funding Editor-in-Chief of the SIAM Journal on Imaging Sciences.

Stéphane Mallat received the Ph.D. degree in electrical engineering from the University of Pennsylvania, Philadelphia, in 1988.

In 1988, he joined the Computer Science Department of the Courant Institue of Mathematical Sciences where he was Associate Professor in 1994 and Professsor in 1996. Since 1995, he has been a full Professor in the Applied Mathematics Department at Ecole Polytechnique, Paris. From 2001 to 2007 he was co-founder and CEO of a semiconductor start-up company.

Dr. Mallat is an IEEE and EURASIP fellow. He received the 1990 IEEE Signal Processing Society's paper award, the 1993 Alfred Sloan fellowship in Mathematics, the 1997 Outstanding Achievement Award from the SPIE Optical Engineering Society, the 1997 Blaise Pascal Prize in applied mathematics from the French Academy of Sciences, the 2004 European IST Grand prize, the 2004 INIST-CNRS prize for most cited French researcher in engineering and computer science, and the 2007 EADS prize of the French Academy of Sciences.

His research interests include computer vision, signal processing and harmonic analysis.

References

[1] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans. on Signal Proc.*, 54(11):4311, 2006.

[2] S. Allassonniere, Y. Amit, and A. Trouvé. Towards a coherent statistical framework for dense deformable template estimation. *J.R. Statist. Soc. B*, 69(1):3–29, 2007.

[3] R.G. Baraniuk, V. Cevher, M.F. Duarte, and C. Hegde. Model-based compressive sensing. *IEEE Trans. on Info. Theo.*, 56:1982–2001, 2010.

[4] J.P. Baudry, A.E. Raftery, G. Celeux, K. Lo, and R. Gottardo. Combining mixture components for clustering. *Journal of Computational and Graphical Statistics*, 19(2):332–353, 2010.

[5] J. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society. Series B (Methodological)*, 48(3):259–302, 1986.

[6] J.M. Bioucas-Dias and M.A.T. Figueiredo. A new TwIST: two-step iterative shrinkage/thresholding algorithms for image restoration. *IEEE Trans. on Image Proc.*, 16(12):2992–3004, 2007.

[7] J.M. Bioucas-Dias, M.A.T. Figueiredo, and J.P. Oliveira. Total variation-based image deconvolution: a majorization-minimization approach. In *ICASSP*, volume 2, 2006.

[8] T. Blu, P. Thévenaz, and M. Unser. MOMS: Maximal-order interpolation of minimal support. *IEEE Trans. on Image Proc.*, 10(7):1069–1080, 2001.

[9] S.P. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[10] A. Buades, B. Coll, and J.M. Morel. A review of image denoising algorithms, with a new one. *Multi. Model. and Sim.*, 4(2):490–530, 2006.

[11] E.J. Candes and D.L. Donoho. New tight frames of curvelets and optimal representations of objects with C2 singularities. *Comm. Pure Appl. Math*, 56:219–266, 2004.

[12] E.J. Candes and T. Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE Trans. on Info. Theo.*, 52(12):5406–5425, 2006.

[13] G. Celeux and G. Govaert. Gaussian parsimonious clustering models. *Pattern Recognition*, 28(5):781–793, 1995.

[14] P. Chatterjee and P. Milanfar. Clustering-based denoising with locally learned dictionaries. *IEEE Trans. on Image Proc.*, 18(7), 2009.

[15] S.S. Chen, D.L. Donoho, and M.A. Saunders. Atomic decomposition by basis pursuit. *SIAM J. on Sci. Comp.*, 20:33, 1999.

[16] Yi Chen and M. Gupta. Em demystified: An expectation-maximization tutorial. *Electrical Engineering*, (206), 2010.

[17] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Trans. on Image Proc.*, 16(8):2080–2095, 2007.

[18] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image restoration by sparse 3D transform-domain collaborative filtering. In *SPIE Electronic Imaging*, 2008.

[19] I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Comm. Pure Appl. Math*, 57:1413–1457, 2004.

[20] I. Daubechies, R. DeVore, M. Fornasier, and S. Güntürk. Iteratively re-weighted least squares minimization for sparse recovery. *Commun. Pure Appl. Math*, page 35, 2009.

[21] A.P. Dempster, N.M. Laird, D.B. Rubin, et al. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[22] D.L. Donoho. Compressed sensing. *IEEE Trans. on Info. Theo.*, 52(4):1289–1306, 2006.

[23] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32(2):407–451, 2004.

[24] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Trans. on Image Proc.*, 15(12):3736–3745, 2006.

[25] M. Elad, J.L. Starck, P. Querre, and D.L. Donoho. Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA). *App. and Comp. Harmonic Analysis*, 19(3):340–358, 2005.

[26] Y.C. Eldar, P. Kuppinger, and H. Bölcskei. Compressed sensing of block-sparse signals: uncertainty relations and efficient recovery. *arXiv: 0906.3173, submitted to IEEE Trans. on Signal Proc.*, 2009.

[27] Y.C. Eldar and M. Mishali. Robust recovery of signals from a union of structured subspaces. *IEEE Trans. Info. Theo.*, 55(11):5302–5316, 2009.

[28] A. Elmoataz, O. Lezoray, and S. Bougleux. Nonlocal discrete regularization on weighted graphs: a framework for image and manifold processing. *Image Processing, IEEE Transactions on*, 17(7):1047–1060, 2008.

[29] M.J. Fadili, J.L. Starck, and F. Murtagh. Inpainting and zooming using sparse representations. *The Comp. J.*, 52(1):64, 2009.

[30] A. Foi, K. Dabov, V. Katkovnik, and K. Egiazarian. Shape-adaptive DCT for denoising and image reconstruction. In *Proceedings of SPIE*, volume 6064, pages 203–214, 2006.

[31] C. Fraley and A.E. Raftery. How many clusters? Which clustering method? Answers via model-based cluster analysis. *The Computer Journal*, 41(8):578–588, 1998.

[32] N. Friedman and S. Russell. Image segmentation in video sequences: a probabilistic approach. In *Uncertainty in Artificial iIntelligence: Proceedings of the Thirteenth Conference*, volume 94720, page 175, 1997.

[33] Z. Ghahramani and M.I. Jordan. Supervised learning from incomplete data via an EM approach. *NIPS*, page 120, 1994.

[34] G. Gilboa and S. Osher. Nonlocal operators with applications to image processing. *Multi. Model. Simul*, 7(3):1005–1028, 2008.

[35] J.A. Guerrero-Colon, L. Mancera, and J. Portilla. Image restoration using space-variant Gaussian scale mixtures in overcomplete pyramids. *IEEE Trans. on Image Proc.*, 17(1):27, 2008.

[36] J.A. Guerrero-Colón, E.P. Simoncelli, and J. Portilla. Image denoising using mixtures of gaussian scale mixtures. In *IEEE ICASSP*, pages 565–568. IEEE, 2008.

[37] O.G. Guleryuz. Nonlinear approximation based image recovery using adaptive sparse reconstructions and iterated denoising–Part II: Adaptive algorithms. *IEEE Trans. on Image Proc.*, 15(3):555–571, 2006.

[38] R.J. Hathaway. Another interpretation of the EM algorithm for mixture distributions. *Stat. & Prob. Letters*, 4(2):53–56, 1986.

[39] J. Huang, T. Zhang, and D. Metaxas. Learning with structured sparsity. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 417–424. ACM, 2009.

[40] R. Jenatton, J.Y. Audibert, and F. Bach. Structured variable selection with sparsity-inducing norms. *arXiv:0904.3523v2*, 2009.

[41] C. Kervrann and J. Boulanger. Local adaptivity to variable smoothness for exemplar-based image regularization and representation. *International Journal of Computer Vision*, 79(1):45–69, 2008.

[42] R. Keys. Cubic convolution interpolation for digital image processing. *IEEE Trans. on Acous, Speech and Signal Proc.*, 29(6):1153–1160, 1981.

[43] F. Léger, G. Yu, and G Sapiro. Efficient Matrix Completion with Gaussian Models. In *ICASSP*, 2011.

[44] A. Levin, R. Fergus, F. Durand, and W.T. Freeman. Deconvolution using natural image priors. *Tech. report, CSAIL/MIT*, 2007.

[45] X. Li. Patch-based image interpolation: algorithms and applications. In *International Workshop on Local and Non-Local Approximation in Image Processing*, 2008.

[46] X. Li and M. T. Orchard. New edge-directed interpolation. *IEEE Trans. on Image Proc.*, 10(10):1521–1527, 2001.

[47] Z. Liang and S. Wang. An EM approach to MAP solution of segmenting tissue mixtures: a numerical analysis. *IEEE Trans. on Medical Imaging*, 28(2):297–310, 2009.

[48] Y. Lou, A.L. Bertozzi, and S. Soatto. Direct sparse deblurring. *J. of Math. Imaging and Vision*, pages 1–12, 2011.

[49] S. Lyu and E.P. Simoncelli. Modeling multiscale subbands of photographic images with fields of gaussian scale mixtures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 693–706, 2009.

[50] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Non-local sparse models for image restoration. In *Proceedings of the IEEE International Conference on Computer Vision*, 2009.

[51] J. Mairal, M. Elad, and G. Sapiro. Sparse representation for color image restoration. *IEEE Trans. on Image Proc.*, 17, 2008.

[52] J. Mairal, G. Sapiro, and M. Elad. Learning multiscale sparse representations for image and video restoration. *SIAM Multiscale Modeling and Simulation*, 7(1):214–241, 2008.

[53] S. Mallat. *A Wavelet Tour of Signal Proc.: The Sparse Way, 3rd edition*. Academic Press, 2008.

[54] S. Mallat and G. Peyré. Orthogonal bandlet bases for geometric images approximation. *Comm. Pure Appl. Math*, 61(9):1173–1212, 2008.

[55] S. Mallat and G. Yu. Super-resolution with sparse mixing estimators. *IEEE Trans. on Image Proc.*, 19(11):2889–2900, 2010.

[56] S.G. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Trans. on Signal Proc.*, 41:3397–3415, 1993.

[57] G.J. McLachlan and K.E. Basford. *Mixture Models: Inference and Applications to Clustering*. New York: Dekker, 1988.

[58] J.M. Morel and G. Yu. On the consistency of the SIFT Method. *Accepted to Inverse problems and Imaging*, 2009.

[59] N. Mueller, Y. Lu, and M.N. Do. Image interpolation using multiscale geometric representations. In *Computational Imaging V. Edited by Bouman, Charles A.; Miller, Eric L.; Pollak, Ilya. Proceedings of the SPIE*, volume 6498, page 64980A, 2007.

[60] A. Muñoz, T. Blu, and M. Unser. Least-squares image resizing using finite differences. *IEEE Trans. on Image Proc.*, 10(9):1365–1378, 2001.

[61] R. Neelamani, H. Choi, and R. Baraniuk. Forward: Fourier-wavelet regularized deconvolution for ill-conditioned systems. *IEEE Trans. on Signal Proc.*, 52(2):418–433, 2004.

[62] H. Permuter, J. Francos, and IH Jermyn. Gaussian mixture models of texture and colour for image database retrieval. In *Proc. ICASSP*, volume 1, pages 25–88, 2003.

[63] G. Peyré, S. Bougleux, and L. Cohen. Non-local regularization of inverse problems. *European Conference on Computer Vision 2008*, pages 57–68.

[64] J. Portilla. Image restoration through l0 analysis-based sparse optimization in tight frames. In *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pages 3909–3912. IEEE, 2009.

[65] J. Portilla, V. Strela, M.J. Wainwright, and E.P. Simoncelli. Image denoising using scale mixtures of Gaussians in the wavelet domain. *IEEE Trans. on Image Proc.*, 12(11):1338–1351, 2003.

[66] S. Roth and M.J. Black. Fields of experts. *International Journal of Computer Vision*, 82(2):205–229, 2009.

[67] J. Schafer and K. Strimmer. A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical Applications in Genetics and Molecular Biology*, 4(1):1175, 2005.

[68] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, 1999.

[69] M. Stojnic, F. Parvaresh, and B. Hassibi. On the reconstruction of block-sparse signals with an optimal number of measurements. *IEEE Trans. on Signal Proc.*, 57(2):3075–3085, 2009.

[70] H. Takeda, S. Farsiu, and P. Milanfar. Kernel regression for image processing and reconstruction. *IEEE Trans. on Image Proc.*, 16(2), 2007.

[71] R. Tibshirani. Regression shrinkage and selection via the lasso. *J. of the Royal Stat. Society*, pages 267–288, 1996.

[72] J. A. Tropp. Greed is good: Algorithmic results for sparse approximation. *IEEE Trans. on Info. Theo.*, 50:2231–2242, 2004.
[73] D. Tschumperlé and L. Brun. Non-local image smoothing by applying anisotropic diffusion pde's in the space of patches. In *IEEE ICIP*, pages 2957–2960. IEEE, 2009.
[74] M. Unser. Splines: A perfect fit for signal and image processing. *IEEE Signal Proc. Magazine*, 16(6):22–38, 1999.
[75] R. Vidal, Y. Ma, and S. Sastry. Generalized principal component analysis (GPCA). *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(12):1945–1959, 2005.
[76] J. Yang, J. Wright, T. Huang, and Y. Ma. Image super-resolution via sparse representation. *Accepted to IEEE Trans. on Image Proc.*, 2010.
[77] G. Yu and Sapiro. G. Statistical compressive sensing with Gaussian models. *submitted, arXiv:1101.5785v1*, Jan. 2011.
[78] G. Yu, Sapiro. G., and S. Mallat. Park City Mathematics Program (PCMI) summer school tutorial. *http://www.cmap.polytechnique.fr/∼yu/research/PLE/talk_PLE_v7.pdf*, 2010.
[79] G. Yu and S. Mallat. Sparse super-resolution with space matching pursuits. *Proc. SPARS09, Saint-Malo*, 2009.
[80] G. Yu, G. Sapiro, and S. Mallat. Image modeling and enhancement via structured sparse model selection. *IEEE ICIP*, 2010.
[81] L. Zhang and X. Wu. An edge-guided image interpolation algorithm via directional filtering and data fusion. *IEEE Trans. on Image Proc.*, 15(8):2226–2238, 2006.
[82] X. Zhang and X. Wu. Image interpolation by adaptive 2-d autoregressive modeling and soft-decision estimation. *IEEE Trans. on Image Proc.*, 17(6):887–896, 2008.
[83] J. Zhou, J.L. Coatrieux, A. Bousse, H. Shu, and L. Luo. A Bayesian MAP-EM algorithm for PET image reconstruction using wavelet transform. *IEEE Trans. on Nuclear Science*, 54(5):1660–1669, 2007.
[84] Zhou, M. and Chen, H. and Paisley, J. and Ren, L. and Li, L. and Xing, Z. and Dunson, D. and Sapiro, G. and Carin, L. Nonparametric bayesian dictionary learning for analysis of noisy and incomplete images. *Image Processing, IEEE Transactions on*, (99), 2011.