SURE Guided Gaussian Mixture Image Denoising

Yi-Qing Wang, Jean-Michel Morel

SIAM J. IMAGING SCIENCES
Vol. xx, pp. x

SURE Guided Gaussian Mixture Image Denoising

Yi-Qing Wang[†] and Jean-Michel Morel[†]

Abstract. The Gaussian mixture is a patch prior that has enjoyed tremendous success in image processing. In this work, by using Gaussian factor modeling, its dedicated Expectation Maximization (EM) inference as well as a statistical filter selection and algorithm stopping rule, we develop SURE (Stein's Unbiased Risk Estimator) guided Piecewise Linear Estimation (S-PLE), a patch-based prior learning algorithm capable of delivering state-of-the-art performance at image denoising. In light of this algorithm's features and its results, we also seek to address the number of components to be included when setting up a Gaussian mixture for image patch modeling. By juxtaposing both options, we show that a simple learned prior can perform as well if not better than a much richer yet fixed prior.

Key words. Gaussian factor mixture, EM algorithm, image denoising, tensor structure, Stein's unbiased risk estimator (SURE)

AMS subject classifications. 68U10, 62H25, 62H30, 94A08

1. Introduction.

A novel patch-based image denoising algorithm, SURE guided Piecewise Linear Estimation (S-PLE), is introduced in this paper[3]. Its underpinning assumption is a classical one: all patches found in an image are generated independently according to a Gaussian Mixture Model (GMM) whereby each component model is roughly responsible for a patch subset characterized by a particular observable feature. Though certainly objectionable on several grounds, the assumption of independence among samples and the use of Gaussian distributions prove extremely popular and are adopted in a wide range of modeling practice, mainly because of the resulting algorithmic tractability and performance.

Before delving into a review of related research and competing algorithms, let us recall that in GMM, an image patch P of size $\kappa \times \kappa$ is postulated to be distributed over $\mathbb{R}^{\kappa^2}$ according to

$$\sum_{k=0}^{K-1} \boldsymbol{w}_k \mathcal{N}(P; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

for some integer K, positive scalars $(\boldsymbol{w}_k)_{0\leq k\leq K-1}$ with $\sum_{k=0}^{K-1} \boldsymbol{w}_k = 1$, vectors $(\boldsymbol{\mu}_k)_{0\leq k\leq K-1}$, and positive semidefinite matrices $(\boldsymbol{\Sigma}_k)_{0\leq k\leq K-1}$ representing the number of Gaussian components in the mixture, their prior probabilities, expectations, and covariance matrices.

`Notational convention:` deterministic parameters required to be estimated for model building are written in bold. And those supposedly random quantities and absolute constants are in normal font.

[2]CMLA, ENS Cachan, 94230 Cachan, France (yqwang9@gmail.com, morel@cmla.ens-cachan.fr). This work was supported in part by the Centre National d'Etudes Spatiales (CNES, MISS Project), the European Research Council (Advanced Grant Twelve Labours), the Office of Naval Research (under Grant N00014-97-1-0839) and the Direction Générale de l'Armement (DGA).

[3]Supporting software and an online demo is available at http://www.ipol.im/pub/pre/52/.

The commonly adopted observation model for image restoration is

$$\tilde{P} = \mathfrak{M}(P + N)$$

which describes the process whereby a patch P undergoes degradation by additive Gaussian noise and linear distortion (for instance, a mask). Under the aforementioned signal generating mechanism, it can be further specialized to

$$\tilde{P} = \mathfrak{M}(\sum_{k=0}^{K-1} 1_{s_P=k} P + N) = \sum_{k=0}^{K-1} \mathfrak{M}(P + N) 1_{s_P=k}$$

with s_P, the patch model selector, a random variable distributed according to $(\boldsymbol{w}_k)_{0\leq k\leq K-1}$ and independent of N. In the context of image denoising, $\mathfrak{M}$ is reduced to the identity and will be removed henceforth.

One can devise a single linear filter to minimize the mean square error (MSE), namely the l^2 distance between the patch P and its estimate on average (in the probability space defined by P and N). Like the very design of Wiener filter, this can be done with

$$(\boldsymbol{L}_*, \boldsymbol{b}_*) = \underset{\boldsymbol{L},\boldsymbol{b}}{\operatorname{argmin}} \mathbb{E}[\|\boldsymbol{L}\tilde{P} + \boldsymbol{b} - P\|_2^2] \tag{1.1}$$

where $(\boldsymbol{L}_*, \boldsymbol{b}_*)$ determines the filter (Appendix A). However, it is generally ill-advised to apply a global linear filter in image processing because of the great variation routinely exhibited by image patches and a fixed linear filter's inability to capture local information. Mathematically speaking, the optimal linear filter's problem formulation imposes too stringent a constraint upon the function space in which to look for the best approximation to the conditional expectation $\mathbb{E}[P|\tilde{P}]$, which, due to the Gaussian mixture prior, actually has a closed form (see Appendix B)

$$\mathbb{E}[P|\tilde{P}] = \sum_{k=0}^{K-1} \mathbb{P}(s_P = k|\tilde{P}) \mathbb{E}[P|s_P = k, \tilde{P}] \tag{1.2}$$

and turns out to be a patch-dependent combination of K fixed linear filters.

Therefore, the key to a good patch-based denoising algorithm, irrespective of modeling assumptions, is a non-linear filter able to adapt its behavior from one patch to another. A review of the progress made in this field over the last few decades was presented by [3], in which Buades, Coll and Morel, inspired in part by the pioneering work [10] of Efros and Leung in texture synthesis, also proposed an effective denoising paradigm named Non-Local Means (NLM) that exploits information redundancy in natural images. Kervrann and Boulanger [14], [15] improved NLM by enabling adaptive smoothing window size selection for similar patch detection. And through similar patch grouping and collaborative filtering, BM3D [6] further enhanced NLM and catapulted it to one of the best-performing denoising methods that define the current state-of-the-art.

Another promising direction of research was initiated in [1], [11] where Elad and Aharon proposed to use a greedy orthogonal matching pursuit algorithm based on the notion that

image patches can be sparsely represented with an over-complete dictionary. In [5], Chatterjee and Milanfar, building on their steering kernel regression framework [27], presented a patch orientation based dictionary learning algorithm which gave rise to the K-LLD denoising algorithm. Yu, Sapiro and Mallat [32] recognized the orientation as the most prominent patch feature and designed a so-called MAP-EM (Maximum A Posteriori Expectation Maximization) algorithm named PLE along a similar vein, but intended as a generic image recovery related inverse problem solver. In a recent development, Zoran and Weiss [33], rather than pursue the best dictionary directly, introduced a new optimisation scheme which continued the effort started as early as in 1992 by Rudin, Osher and Fatemi [24] of seeking an adequate description of image priors. Instead of constructing priors for images as a whole as did their predecessors, they focused on a prior for image patches in the form of a Gaussian mixture and obtained impressive results.

In this paper, motivated in part by the mentioned work [5], [32], [33], we present S-PLE which, unlike those methods spawned by the NLM paradigm, groups image patches by assessing patch-to-model rather than patch-to-patch distance. As will be clear from our discussion in Section 3, the ability of patches to choose among filters and adapt their own forms and sizes to image content and noise level is of critical importance to both visual quality and accuracy of the restoration. S-PLE addresses the issue by using SURE [26] as a decision aid which enables the desired adaptive filtering and results in a state-of-the-art performance in terms of MSE, thereby representing a substantial improvement over the existing algorithms such as K-LLD and PLE in the same category. In addition, thanks to SURE, we show how to track S-PLE's real-time performance with a simple device. Finally, although the Gaussian mixture is widely recognized as an effective tool for probability density approximation, opinions differ as to the number of components a mixture should have for high-dimensional data modeling in the patch space. In light of the results produced by S-PLE, we show that a good adaptive prior made up of a small number of Gaussian components can do just as well, if not better than a much richer yet fixed prior.

The rest of the paper is organized as follows: Section 2 gives a summary of PLE [32] and points out its major shortcomings with respect to S-PLE. A detailed account of S-PLE is then provided in Section 3. Section 4 presents the new algorithm outline and a comparative study. Section 5 addresses the number of components to be included in a Gaussian mixture for image patch modeling. The last section is devoted to deriving the two-stage EM algorithm for Gaussian factor mixture inference as well as other stated results.

2. PLE. In this section, PLE is summarized to highlight its difference with S-PLE. In [32], the authors start with $K-1$ directional models derived from synthetic samples and one additional Gaussian model using DCT as its basis. For each patch from an image to restore, PLE then produces K estimates under these model assumptions and retains the one that maximizes the conditional probability to have both the observation and its estimate. Finally, the parameters of all these models are updated with the obtained estimates. The last two steps, referred to as estimation and maximization in the paper, are then repeated several times before the algorithm terminates.

[4]The implemented PLE for our comparative study leaves out both component substitution and basis orthogonalization because they can cause numerical instability as it is difficult to tell whether a set of vectors

Algorithm 1 Initialization of PLE

Parameter: Number of Gaussian models K, patch dimension $\kappa \times \kappa$.
for $k = 0$ to $K - 2$ **do**
Create and sample synthetic images
1. Create a binary image B of size 100×100 taking value in $\{0, 255\}$ with two sets of pixels $\{(r, u) : \ B(r, u) = 0\}$ and $\{(r, u) : \ B(r, u) = 255\}$ separated by a straight line which passes through the center of the image with its normal inclined at $\frac{k}{K-1}\pi$.
2. Blur B with Gaussian kernels of different standard deviations $(\sigma_b)_{1\le b\le 4}$ with $\sigma_b = 2b$ for all b.
3. Sample a large number of $\kappa \times \kappa$ patches from these blurred images to form the patch set $\mathcal{P}_k$.

Compute the statistics
1. Estimate the model mean and covariance:

$$\boldsymbol{\mu}_k = \frac{1}{|\mathcal{P}_k|} \sum_{P\in\mathcal{P}_k} P, \quad \boldsymbol{\Sigma}_k = \frac{1}{|\mathcal{P}_k|} \sum_{P\in\mathcal{P}_k} (P - \boldsymbol{\mu}_k)(P - \boldsymbol{\mu}_k)^T.$$

2. Set the k-th directional basis V_k using the spectral decomposition $\boldsymbol{\Sigma}_k = V_k \Lambda_k V_k^T$.
3. Set $\boldsymbol{\mu}_k = 0$. Replace the first leading eigenvector in V_k by a normalized DC component and apply Gram-Schmitt to orthogonalize the remaining vectors.[4]

end for
Add DCT (with its component frequencies arranged in ascending order) to this setup as its $(K - 1)$-th model basis. Set the $(K - 1)$-th model mean to zero.
Take a sequence of κ^2 positive numbers of exponential decay (a working example: $m \in [0, \kappa^2 - 1] \cap \mathbb{Z} \mapsto 2^{20.5-0.5m}$) and make them the eigenvalues of all K Gaussian models.

Comments on the algorithm.

(a) PLE is not GMM-based for it involves no concept of model priors. Put in another way, if there is to be a mixture in PLE, all its components must have equal mixing weights all the time: although not explicitly stated, Laplace's principle of insufficient reason is invoked while it is not required.

(b) Its directional basis initialization literally cries out for factor loading matrix reduction as explained in the next section while it is not taken into account. Its DCT basis is supposed to handle both textural and flat patches, which leads to artifacts in recovered images.

(c) No criterion is guaranteed to converge in any sense, making it difficult to decide when to stop the algorithm. Likewise, the way the parameters are estimated is questionable: because of the preceding filtering, the estimators produced in the maximization step are likely to be biased, which could have been avoided by using noisy patches directly rather than their estimates.

(d) The algorithm is rigid and cannot alter its behavior in response to noise level. And there is too much latitude in tuning parameters such as the model eigenvalues, which often undermines the algorithm's numerical stability and performance.

are collinear with the computer's limited precision. With DC components removed from the directional bases, PLE could discriminate better.

Algorithm 2 PLE

Input: A noisy gray image $\tilde{U}$, its noise standard deviation σ and the initial setup Θ_0.
Parameter: Number of PLE iterations S.
Read in Θ_0. Extract all $\kappa \times \kappa$ patches from $\tilde{U}$ to have $(\tilde{P}_i)_{1\le i\le N}$.
for $t = 1$ to S **do**
Estimation:
1. Maximize the conditional density given the observation and the model:

$$\begin{aligned}\forall (i,k),\quad \widehat{P}_i^{(k)} &= \operatorname*{argmax}_P p(P|\tilde{P}_i, \boldsymbol{\mu}_{k,t-1}, \boldsymbol{\Sigma}_{k,t-1}) \\ &= \operatorname*{argmax}_P p(P, \tilde{P}_i|\boldsymbol{\mu}_{k,t-1}, \boldsymbol{\Sigma}_{k,t-1}) \\ &= \operatorname*{argmin}_P \Big(\frac{\|P-\tilde{P}_i\|^2}{\sigma^2} + (P-\boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_{k,t-1}^{-1}(P-\boldsymbol{\mu}_k)\Big).\end{aligned}$$

2. Select the model that best fits the i-th observation and its conditional estimate:

$$\begin{aligned}k_i &= \operatorname*{argmax}_{0\le k\le K-1} p(\widehat{P}_i^{(k)}, \tilde{P}_i|\boldsymbol{\mu}_{k,t-1}, \boldsymbol{\Sigma}_{k,t-1}) \\ &= \operatorname*{argmin}_{0\le k\le K-1} \Big(\frac{\|\widehat{P}_i^{(k)}-\tilde{P}_i\|^2}{\sigma^2} + (\widehat{P}_i^{(k)}-\boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_{k,t-1}^{-1}(\widehat{P}_i^{(k)}-\boldsymbol{\mu}_k) + \ln\det\boldsymbol{\Sigma}_{k,t-1}\Big)\end{aligned}$$

which leads to the estimated patch $\widehat{P}_i = \widehat{P}_i^{(k_i)}$ and its assignment to the k_i-th model.
Maximization: Denote $\mathcal{Q}_k$ the set of estimated patches attributed to the k-th model.
for $k = 0$ to $K-1$ **do**
Estimate the model mean and covariance:

$$\boldsymbol{\mu}_{k,t} = \frac{1}{|\mathcal{Q}_k|}\sum_{P\in\mathcal{Q}_k} P, \quad \boldsymbol{\Sigma}_{k,t} = \frac{1}{|\mathcal{Q}_k|}\sum_{P\in\mathcal{Q}_k}(P-\boldsymbol{\mu}_{k,t})(P-\boldsymbol{\mu}_{k,t})^T + \epsilon I$$

where ϵ is a small positive number to ensure the definiteness of $\boldsymbol{\Sigma}_{k,t}$.
end for
end for
Assign equal weights to all restored patches and recover the image.

(e) Certain choices made in the initialization step are hard to interpret. For example, the vanishing model means and the rectification of the first leading eigenvector in all directional bases. Perhaps they are there to prevent computational singularities resulting from the potentially ill-chosen parameters such as the model eigenvalues.

3. SURE Guided PLE Denoising Algorithm.

3.1. Gaussian Factor Mixture. The covariance matrix for patches of a particular orientation θ, modeled under the Gaussian assumption, ought not to be of full rank. Indeed, if it were, all its eigenvectors, viewed as individual patches, would be oriented in either θ or $\pi-\theta$. As a result, none of their linear combinations can produce a patch with an orientation equal

to $\frac{\pi}{2} - \theta$, which contradicts the very definition of full rank.

However, instead of just θ, when building a model, one usually lets it account for a slightly wider range of orientations, i.e. $(\theta - \Delta, \theta + \Delta)$ with a small scalar $\Delta > 0$. Thus, the above reasoning need not apply. There is still reason to believe though that a reduced set of vectors suffices to represent patches of a narrow range of orientations. Therefore instead of a full-fledged Gaussian distribution, an equally flexible and yet more appropriate candidate in this case is a Gaussian factor model (GFM)

$$P_\theta = \boldsymbol{F}_\theta c + \boldsymbol{\mu}_\theta$$

where patch variability can be restricted by limiting the number of columns l contained in the factor loading matrix $\boldsymbol{F}_\theta \in \mathbb{R}^{\kappa^2 \times l}$. With $\boldsymbol{\mu}_\theta$ deterministic and c following the Gaussian law $\mathcal{N}(0, I_l)$, P_θ remains Gaussian. Moreover, the factor trimming results in a more regularized model behavior which can help resist over-fitting.

In addition to a number of *mono-oriented* models, two more *non-oriented* components for textural and flat patches respectively should be set up to complete the mixture.

To sum up, the i-th noisy patch is assumed by S-PLE to follow:

$$\tilde{P}_i = \sum_{k=0}^{K-1} (\boldsymbol{F}_k c_i + \boldsymbol{\mu}_k + \boldsymbol{\sigma} n_i) 1_{s_i = k}$$

where

1. $\boldsymbol{F}_k \in \mathbb{R}^{\kappa^2 \times l_k}$: a deterministic matrix containing l_k factors used by the k-th model;
2. $c_i \in \mathbb{R}^{l_k}$: a Gaussian coefficient distributed as $\mathcal{N}(0, I_{l_k})$;
3. $\boldsymbol{\mu}_k \in \mathbb{R}^{\kappa^2}$: a deterministic vector representing the k-th model mean;
4. $\boldsymbol{\sigma} \in \mathbb{R}_+$: the standard deviation of some zero-mean additive Gaussian noise;
5. $n_i \in \mathbb{R}^{\kappa^2}$: a Gaussian vector following $\mathcal{N}(0, I_{\kappa^2})$ independent of c_i;
6. $s_i \in \{0, \cdots, K-1\}$: a discrete random variable that selects a model for the i-th patch.

When it comes to learning the hidden parameters of a mixture from an observed dataset, the renowned Expectation Maximization [7] is arguably the algorithm of choice. A variant dedicated to the GFM mixture inference has been developed by Tipping and Bishop [28] of which a detailed account can be found in Appendix C.

3.2. GFM Mixture Initialization. For EM to succeed at its task, a good starting point is key in that the algorithm can be trapped at local maxima and consequently fail to reach global maxima. Synthetic image sampling suggested in [32], though interesting, does not allow the construction of an appropriate prior for lack of information to estimate the mixing weights. A more reasonable solution is to draw samples directly from natural images with the help of the so-called "tensor structure" orientation detector [12]: given a square patch P, the discrete gradient $\nabla P(r, u)$ is computed at all pixel sites in its domain $Dom(P)$. Then the patch's

orientation v_* is found by

$$\begin{aligned} v_* &= \underset{\|v\|=1}{\operatorname{argmin}} \sum_{(r,u)\in Dom(P)} \|\nabla P(r,u) - \langle v, \nabla P(r,u)\rangle v\|^2 \\ &= \underset{\|v\|=1}{\operatorname{argmin}} \sum_{(r,u)\in Dom(P)} \|\nabla P(r,u)\|^2 - \langle v, \nabla P(r,u)\rangle^2 \\ &= \underset{\|v\|=1}{\operatorname{argmax}}\, v^T \Big(\sum_{(r,u)\in Dom(P)} \nabla P(r,u)(\nabla P(r,u))^T \Big) v \end{aligned}$$

where $\langle \cdot, \cdot \rangle$ denotes the scalar product between two 2×1 column vectors valued in $\mathbb{R}^2$, The problem is easily solved by computing the first leading eigenvector of the positive semidefinite matrix enclosed in the parentheses, denoted henceforth by M_P. Because of the equality

$$\sum_{(r,u)\in Dom(P)} \|\nabla P(r,u)\|^2 = tr(M_P) = \lambda_s + \lambda_b,$$

it seems natural to declare P oriented if the criterion

$$\frac{\sum_{(r,u)\in Dom(P)} \|\nabla P(r,u) - \langle v_*, \nabla P(r,u)\rangle v_*\|^2}{\sum_{(r,u)\in Dom(P)} \|\nabla P(r,u)\|^2} = \frac{\lambda_s}{\lambda_s + \lambda_b}$$

is small, where λ_b and λ_s ($\lambda_b \geq \lambda_s \geq 0$) are two eigenvalues of M_P. Thus a threshold $t_{\mathbf{orient}} = 5$ is tuned according to our subjective view so that a patch satisfying $\lambda_s^{-1}\lambda_b \geq t_{\mathbf{orient}}$ is likely to be seen as oriented. Its orientation θ_* can then be set to $\psi(\arctan \frac{y_*}{x_*})$ with $v_* = (x_*, y_*)^T$ and $\psi(a) = a1_{a\geq 0} + (\pi + a)1_{a<0}$, the latter of which ensures the positivity of θ_*.

To distinguish between two categories of non-oriented patches, one applies the following rule

$$\lambda_b \geq t_{\mathbf{flat}} \quad \text{and} \quad \lambda_s^{-1}\lambda_b < t_{\mathbf{orient}}$$

as an empirical definition of *multi-oriented* (or textural) patches ($t_{\mathbf{flat}} = 10^4$). The remaining set of patches satisfying

$$\lambda_b < t_{\mathbf{flat}} \quad \text{and} \quad \lambda_s^{-1}\lambda_b < t_{\mathbf{orient}}$$

are seen as *essentially flat*.

The previous definitions split the first quadrant $(\lambda_s, \lambda_b) \in \mathbb{R}_+^2$ into three regions, among which the one characterized by $\lambda_s^{-1}\lambda_b \geq t_{\mathbf{orient}}$ will be further divided into $K-2$ sub-areas by angle quantification to form a K-zone partition. The way to achieve this is to assign a patch P to the k-th mono-oriented model if and only if it satisfies

$$\lambda_s^{-1}(P)\lambda_b(P) \geq t_{\mathbf{orient}} \quad \text{and} \quad \theta_*(P) \in [\frac{k}{K-2}\pi, \frac{k+1}{K-2}\pi)$$

where the notations $\lambda_s(P), \lambda_b(P)$ and $\theta_*(P)$ are meant to emphasize their dependences on P.[5]

Algorithm 3 GMM initialization of S-PLE

Input: Z noiseless natural gray images.
Parameter: Number of mixture components K, patch dimension $\kappa \times \kappa$.
For all $0 \leq k \leq K-1$, set N_k, the number of samples obtained for the k-th model, to 0.
Collect samples:
while $\min_{0\leq k\leq K-1} N_k < 5000$ **do**
Randomly picks one among Z images and sample a $\kappa \times \kappa$ patch P from it.
Calculate the eigenvalues (λ_b, λ_s) of $\sum_{(r,u)\in Dom(P)} \nabla P(r,u)(\nabla P(r,u))^T$ together with its eigenvector v associated with λ_b ($\lambda_b \geq \lambda_s$) where $\nabla P(r,u)$ represents the discrete gradient of P at (r,u).
if $\lambda_b/\lambda_s < t_{\textbf{orient}}$ **then**
if $\lambda_b < t_{\textbf{flat}}$ **then**
Assign P to the flat model: $N_{K-1} \leftarrow N_{K-1} + 1$.
else
Assign P to the multi-oriented model: $N_{K-2} \leftarrow N_{K-2} + 1$.
end if
else
Determine the orientation $\theta = \psi(\arctan \frac{y}{x})$ with $v = (x,y)^T$ and $\psi(a) = a1_{a\geq 0} + (\pi + a)1_{a<0}$.
Assign P to the k-th mono-oriented model if $\theta \in [\frac{k}{K-2}\pi, \frac{k+1}{K-2}\pi)$: $N_k \leftarrow N_k + 1$.
end if
end while
Compute the statistics:
for $k = 0$ to $K-1$ **do**
Estimate the model prior: $\boldsymbol{w}_k = \frac{N_k}{\sum_{j=0}^{K-1} N_j}$.
Estimate the model mean and covariance: denote $\mathcal{P}_k$ the set of patches attributed to the k-th model

$$\boldsymbol{\mu}_k = \frac{1}{|\mathcal{P}_k|} \sum_{P\in\mathcal{P}_k} P, \quad \boldsymbol{\Sigma}_k = \frac{1}{|\mathcal{P}_k|} \sum_{P\in\mathcal{P}_k} (P - \boldsymbol{\mu}_k)(P - \boldsymbol{\mu}_k)^T.$$

Estimate the factor loading matrix: denote l_k the number of factors required by the k-th model. The spectral decomposition $\boldsymbol{\Sigma}_k = V\Lambda V^T$ with $V = [\phi_1, \cdots, \phi_{\kappa^2}]$ and $\Lambda = diag(\lambda_1, \cdots, \lambda_{\kappa^2})$ gives

$$\boldsymbol{F}_k = [\ (\lambda_1 - \sigma^2)^{1/2}\phi_1, \cdots, (\lambda_{l_k} - \sigma^2)^{1/2}\phi_{l_k}\ ]$$
$$\text{where} \quad \sigma^2 = \frac{1}{\kappa^2 - l_k} \sum_{m=l_k+1}^{\kappa^2} \lambda_m.$$

end for

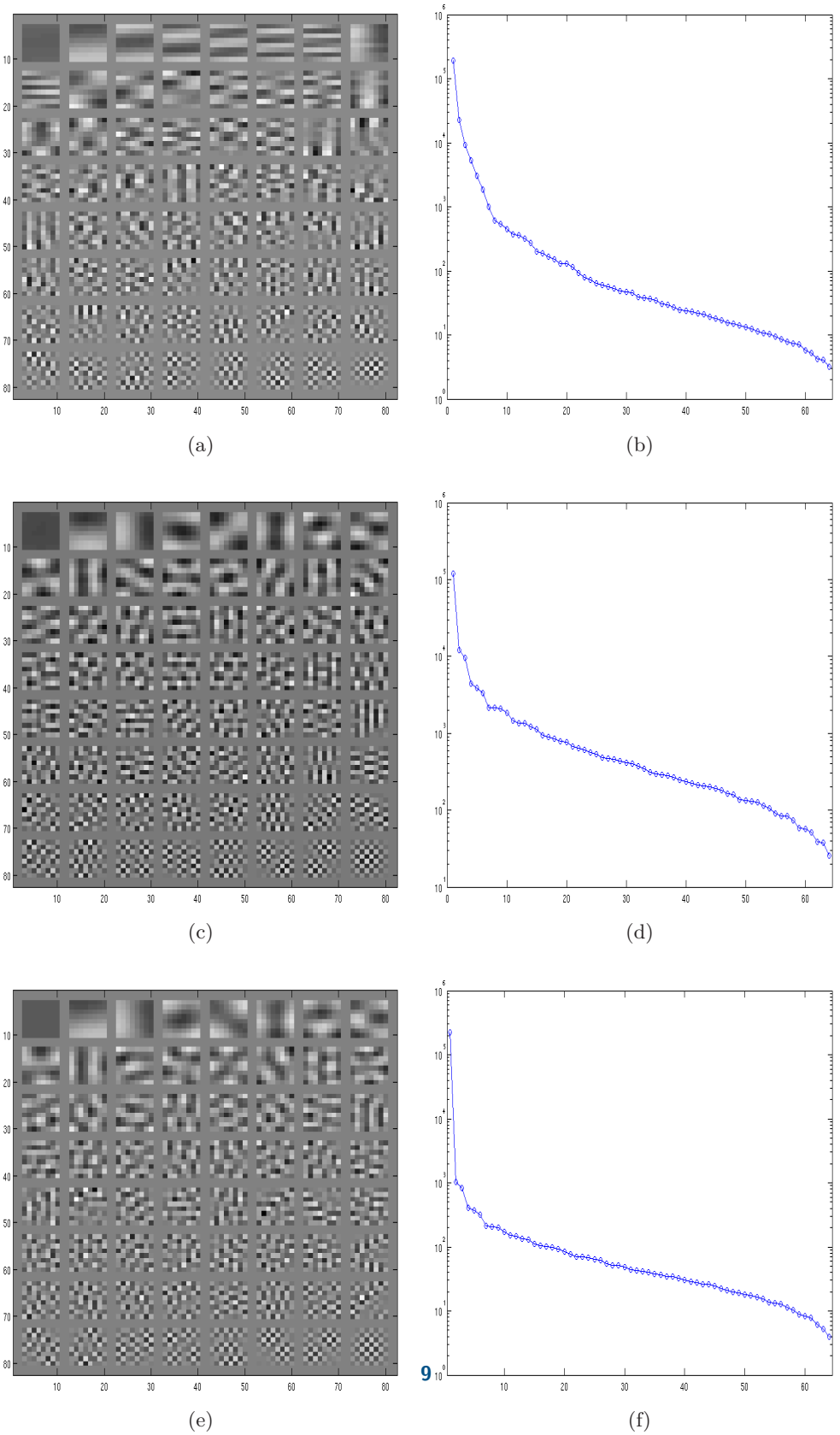


Figure 3.1. *Examples of the eigenvectors and eigenvalues obtained by sampling* 493 *gray images from The Berkeley Segmentation Dataset with* $K = 20$*: a patch view of the eigenvectors of the (a)* 0*-th (mono-oriented), (c)* 18*-th (multi-oriented), (e)* 19*-th (flat) cluster and their eigenvalues displayed in the logarithmic scale: the (b)* 0*-th, (d)* 18*-th and (f)* 19*-th cluster.*

We set $K = 20$ and collected for each model a minimum of 5000 8×8 patches by randomly sampling 493 gray natural images from *The Berkeley Segmentation Dataset*. Shown in Fig.3.1 are three resulting covariance matrices. Several observations are in order:

(a) first, leading eigenvectors in the oriented models preserve their model feature orientation and suggest that low frequency patterns tend to appear more often in natural scenes[6];

(b) second, due to the imprecise nature of orientation definition and measurement, the obtained oriented models' eigenvalues do not go to zero as projected by GFM. However, their still rapid decline in value does not deviate far from what is expected of the model either. Hence it seems reasonable to keep the first few (e.g. 32) factors and reject the rest;

(c) third, contrary to what is observed in the mono-oriented models, the multi-oriented model's eigenvectors bear striking resemblance to those of the flat model. Rather, it is the behavior of the associated eigenvalues that allows us to tell them apart, a hardly surprising fact considering the way we defined and collected them. To prevent over-fitting, the first leading eigenvector was made the sole factor representing the flat model, a convenient practice only possible because of the adoption of the GFM framework;

(d) finally, although the multi-oriented model's eigenvectors do not form DCT, the two bases do look alike, thereby lending support to the choice made in [32]. As a reflection of the richness of textural content, the number of components in the multi-oriented model was set to 63: a DCT-like isotropic basis is thus broken up into two to handle two radically different patch categories.

The spectral decomposition $\mathbf{\Sigma} = V\Lambda V^T$ can be applied to retrieve the desired factors

$$\boldsymbol{F} = [\ \lambda_1^{1/2}\phi_1, \cdots, \lambda_l^{1/2}\phi_l\ ]$$

where ϕ_m (resp. λ_m) is the m-th column of the unitary matrix V (resp. m-th diagonal entry of the diagonal matrix Λ). However, since ultimately a GFM mixture will be used to fit the observation, it can be argued that we do the same at this stage. We thus look for an element in $\mathfrak{C} = \{\boldsymbol{F}\boldsymbol{F}^T + \boldsymbol{\sigma}^2 I, \boldsymbol{F} \in \mathbb{R}^{\kappa^2 \times l}, \boldsymbol{\sigma} \in \mathbb{R}_+\}$ that achieves the highest empirical likelihood for the i.i.d. samples:

$$\begin{aligned}
(\boldsymbol{F}_*, \boldsymbol{\sigma}_*) &= \underset{\boldsymbol{F}\in\mathbb{R}^{\kappa^2\times l}, \boldsymbol{\sigma}\in\mathbb{R}_+}{\operatorname{argmax}} \log \prod_{i=1}^{N} \frac{\exp\Big(-\frac{1}{2}(P_i-\mu)^T(\boldsymbol{F}^T\boldsymbol{F}+\boldsymbol{\sigma}^2 I)^{-1}(P_i-\mu)\Big)}{\sqrt{(2\pi)^{\kappa^2}\det(\boldsymbol{F}^T\boldsymbol{F}+\boldsymbol{\sigma}^2 I)}} \\
&= \underset{\boldsymbol{F}\in\mathbb{R}^{\kappa^2\times l}, \boldsymbol{\sigma}\in\mathbb{R}_+}{\operatorname{argmin}} \frac{N}{2}\Big[\log\det(\boldsymbol{F}^T\boldsymbol{F}+\boldsymbol{\sigma}^2 I) + \frac{1}{N}\sum_{i=1}^{N} tr\Big((\boldsymbol{F}^T\boldsymbol{F}+\boldsymbol{\sigma}^2 I)^{-1}(P_i-\mu)(P_i-\mu)^T\Big)\Big] \\
&= \underset{\boldsymbol{F}\in\mathbb{R}^{\kappa^2\times l}, \boldsymbol{\sigma}\in\mathbb{R}_+}{\operatorname{argmin}} \log\det(\boldsymbol{F}^T\boldsymbol{F}+\boldsymbol{\sigma}^2 I) + tr\Big((\boldsymbol{F}^T\boldsymbol{F}+\boldsymbol{\sigma}^2 I)^{-1}\Sigma\Big)
\end{aligned}$$

which is equivalent to minimizing the Kullback-Leibler divergence between two multivariate Gaussian distributions $\mathcal{N}(\mu, \Sigma)$ and $\mathcal{N}(\mu, \boldsymbol{F}^T\boldsymbol{F} + \boldsymbol{\sigma}^2 I)$ (see Appendix D). The problem has

[5]The patch space is now endowed with a coarse σ-algebra entirely induced by the tensor structure.

[6]If it exists, the *true* probability on the patch space endowed with the Kolmogorov σ-algebra assigns a predominant weight to low frequency patches, in view of their much bigger variance.

been dealt with and lead to probabilistic PCA, a probabilistic interpretation of Principal Component Analysis [28], [29], [23]. Its solution turns out to be quite intuitive

$$\boldsymbol{F}_* = [\ (\lambda_1 - \boldsymbol{\sigma}^2)^{1/2}\phi_1, \cdots, (\lambda_l - \boldsymbol{\sigma}^2)^{1/2}\phi_l\ ]$$
$$\boldsymbol{\sigma}^2 = \frac{1}{\kappa^2 - l}\sum_{m=l+1}^{\kappa^2} \lambda_m$$

with the factor loading matrix $\boldsymbol{F}_*$ unique up to a rotation in the l-dimensional space, which is of no concern to us because the Gaussian measure $\mathcal{N}(0, I_l)$ is rotation-invariant. Note that a restriction on the number of factors in $\mathbf{F}$ is required for regularization purpose: with more factors allowed entry, $\mathfrak{C}$ gets closer to the set of all covariance matrices of dimension $\kappa^2 \times \kappa^2$ and the optimisation will eventually lose its relevance because its solution will no longer be unique in the sense just stated.

In addition to individual model configurations, the patch sampling revealed yet another valuable piece of information regarding the initial mixture structure, namely $(\boldsymbol{w}_k)_{0\le k\le K-1}$, the prior probability of having a randomly selected patch belonging to a particular model. It was estimated by

$$\boldsymbol{w}_k = \frac{N_k}{\sum_{j=0}^{K-1} N_j}$$

with N_k the total number of patches collected for the k-th model satisfying $\min_{0\le k\le K-1} N_k = 5000$. As shown in Fig.3.2(c), the mixing weights can be configured in such a way that non-oriented patches are much more likely to appear than their mono-oriented counterparts. Moreover, within the non-oriented category, the essentially flat patches were made to have a slightly higher probability to show up. This setup conveys our prior belief on the patch composition of a typical natural image and is not image specific: the mixture prior so produced was used in all our experiments.

3.3. Patch Classification with EM. Here we present a concrete example in the hope of better illustrating EM's effectiveness at classifying noisy patches. To this end, we took a color image `dice` from the `IPOL` (*Image Processing On Line*) website and added to its color channels some simulated i.i.d. zero-mean Gaussian noise with standard deviation equal to 10.

To three color components $(u_{\mathbf{R}}, u_{\mathbf{G}}, u_{\mathbf{B}})$, we applied the next luminance-chrominance transformation intended to increase the first transformed channel's signal-to-noise (SNR) ratio:

$$\tilde{u}_1 = \frac{u_{\mathbf{R}} + u_{\mathbf{G}} + u_{\mathbf{B}}}{3}$$
$$\tilde{u}_2 = \frac{u_{\mathbf{R}} - u_{\mathbf{B}}}{\sqrt{2}}$$
$$\tilde{u}_3 = \frac{u_{\mathbf{R}} - 2u_{\mathbf{G}} + u_{\mathbf{B}}}{\sqrt{6}}.$$

To be consistent with the origin (gray images) of the collected statistics, the denominator in the first transformation was set to 3 instead of noise normalizing $\sqrt{3}$ because these components

are believed to be highly correlated. Since a commonly used RGB to gray conversion does not assign equal weight to individual channels (see for instance Matlab's routine `rgb2gray`), one should perhaps have sampled color images accordingly in the first place. But this reasoning implies the necessity to set up four different initial mixtures, one for gray images and the other three for color images which should also depend on the luminance-chrominance transformation chosen by the algorithm. Here we did not make the distinction just to keep the matter simple.

20 models, each containing 32 factors except for the two non-oriented ones, were read in to help set up the initial prior. With noise standard deviation set to $10/\sqrt{3}$, we ran EM on $\tilde{u}_1$. At the end of each iteration, there was for every observed noisy patch $\tilde{P}$ a set of newly calculated posterior probabilities $\{\mathbb{P}(s_P = k \mid \tilde{P}),\ 0 \le k \le 19\}$, which allowed us to determine the most suitable model for $\tilde{P}$ simply by comparing its likelihoods under different model assumptions:

$$k^* = \underset{0\le k\le 19}{\operatorname{argmax}}\, \mathbb{P}(s_P = k \mid \tilde{P}) = \underset{0\le k\le 19}{\operatorname{argmax}}\, \mathbb{P}(\tilde{P} \mid s_P = k)\mathbb{P}(s_P = k).$$

It should be clear by now that updating the mixing weights at the same time as the model parameters is not only required to keep the overall likelihood increasing as the algorithm iterates on, it also helps reduce the misclassification risk and artifacts: for instance, in an image with predominant presence of flat patches, a patch should be assigned to a mono-oriented model only if there is a compelling enough indication to justify the action.

A patch-to-model mapping, henceforth referred to as the *patch map*, can be formed by associating to each patch its most probable model. In the present example, the patch map (Fig.3.2) shows that by the time the first EM iteration ended, pretty much as expected, an overwhelming majority (87.4%) of patches identified with the flat model, thereby preparing the ground for the denoising algorithm's next stage: adaptive filtering.

3.4. SURE-Aided Adaptive Filtering. As pointed out in the introduction, it is crucial for a filter to adapt its behavior to the patch it deals with. Since the ultimate denoising is routinely performed by Wiener or shrinkage filters [9], [8] and that they necessitate a projection operation which consists of expressing an observed patch as a linear combination of the vectors contained in a prefixed basis, the filter's adaptability is translated into its ability to produce vectors that allow sparse coding for any given patch. If this condition proves too difficult to satisfy, it would be desirable if that effort could be directed at the patch categories which tend to appear frequently and are relatively easy to characterize.

Our EM-enabled patch orientation based paradigm suits this purpose well. Because of the created patch map as well as the evolving mixture parameters, EM, as we have seen, turns out to be a convenient tool at building and selecting the best basis among 20 alternatives for patch representation and thus denoising. For example, if a noisy patch $\tilde{P}$ is found to be best described by the k-th model

$$\tilde{P} = \boldsymbol{F}_k c + \boldsymbol{\mu}_k + \boldsymbol{\sigma} N$$

a reasonable basis for its representation will be the one formed by the eigenvectors of $\boldsymbol{F}_k \boldsymbol{F}_k^T$.

Our strategy is to retain a noisy image and keep on updating the GFM mixture as well as the ensuing adaptive filters for individual patches. Consequently the blurring is less an issue than in [5]. On the other hand, although the constantly increasing overall likelihood is an

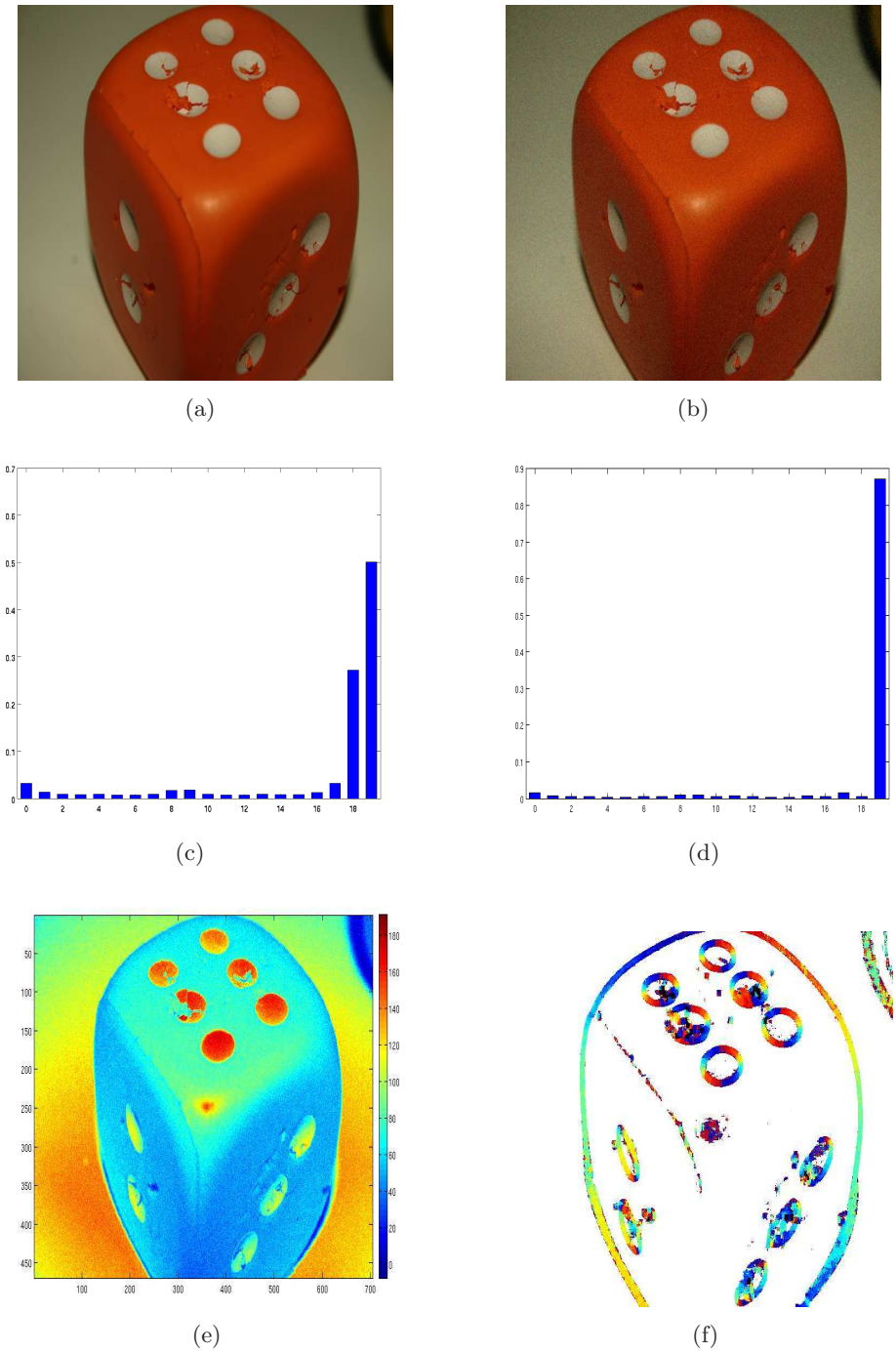


Figure 3.2. *(a) original image (b) noisy image ($\sigma = 10$) (c) initial model priors (d) updated model priors after the first EM iteration on the first transformed channel (e) pixel-wise arithmetic mean of the noisy image's three color channels (f) patch map formed after the first EM iteration. Each color represents a model and white color highlights the patches classified as essentially flat.*

attractive property of the EM algorithm, it does not guarantee monotone convergence of the estimates, except in some special cases [13]. As a matter of fact, notwithstanding the observed tendency for a higher overall likelihood to go with a lower MSE, no causal relationship between the two can be established empirically. Zoran and Weiss [33] attempted to reconcile these two concerns by tying them together to form a single cost function. And we will address the problem with the help of SURE [26] by evaluating a statistic indicative of the adaptive filter's real-time performance. Let us state a specialized version of Stein's theorem in anticipation of its application in this context.

Definition 3.1. *Let $\tilde{P}$ be the sum of a fixed vector $P \in \mathbb{R}^{\kappa^2}$ and a Gaussian random vector $\sigma N \in \mathbb{R}^{\kappa^2}$ with N distributed as $\mathcal{N}(0, I_{\kappa^2})$ and σ a scalar. Let f be a filter of one of the following three forms:*

1. *linear:* $f(\tilde{P}) = \sum_{j=1}^{\kappa^2} c_j \langle \tilde{P} - \mu, b_j \rangle + \mu$
2. *soft shrinkage:* $f(\tilde{P}) = \sum_{j=1}^{\kappa^2} c_j \gamma_t^{\boldsymbol{soft}}(\langle \tilde{P} - \mu, b_j \rangle) + \mu$ *with* $\gamma_t^{\boldsymbol{soft}}(\omega) = sgn(\omega)(|\omega| - t)$
3. *hard shrinkage:* $f(\tilde{P}) = \sum_{j=1}^{\kappa^2} c_j \gamma_t^{\boldsymbol{hard}}(\langle \tilde{P} - \mu, b_j \rangle) + \mu$ *with* $\gamma_t^{\boldsymbol{hard}}(\omega) = \omega 1_{|\omega|>t}$

where $\mu, (c_j)_{1\le j\le\kappa^2}, (b_j)_{1\le j\le\kappa^2}$, and t denote the filter-specific mean, filtering coefficients, basis, and threshold. And their weak derivatives are defined to be

1. *linear:* $\nabla \cdot f(\tilde{P}) = \sum_{j=1}^{\kappa^2} c_j$
2. *soft shrinkage:* $\nabla \cdot f(\tilde{P}) = \sum_{j=1}^{\kappa^2} c_j 1_{[t,+\infty)}(|\langle \tilde{P} - \mu, b_j \rangle|)$
3. *hard shrinkage:* $\nabla \cdot f(\tilde{P}) = \sum_{j=1}^{\kappa^2} c_j \big(1_{[t,+\infty)}(|\langle \tilde{P} - \mu, b_j \rangle|) + t\mathbb{E}\big[(\delta_t - \delta_{-t})(\langle \tilde{P} - \mu, b_j \rangle) \mid P\big]\big)$

where $\delta_x(\cdot)$ represents a Dirac centered on $x \in \mathbb{R}$.

Theorem 3.2. *Under the assumptions in Definition 3.1, SURE given the observation $\tilde{P}$*

$$SURE_f(\tilde{P}) := \frac{1}{\kappa^2}\|\tilde{P} - f(\tilde{P})\|^2 - \sigma^2 + \frac{2\sigma^2}{\kappa^2}\nabla \cdot f(\tilde{P})$$

is unbiased

$$\mathbb{E}[SURE_f(\tilde{P}) \mid P] = \mathbb{E}[\frac{1}{\kappa^2}\|P - f(\tilde{P})\|^2 \mid P].$$

SURE is valuable because it is a function of only the observable $\tilde{P}$. However, in case of f being a hard shrinkage operator, the expectation evaluating the Gaussian density difference at t and $-t$

$$\mathbb{E}\big[(\delta_t - \delta_{-t})(\langle \tilde{P} - \mu, b_j \rangle) \mid P\big]$$

is a function of the unknown P. To circumvent the issue, one only needs to replace the expectation with an approximatively unbiased estimator

$$\frac{1}{2\epsilon}(1_{[t-\epsilon,t+\epsilon]} - 1_{[-t-\epsilon,-t+\epsilon]})(\langle \tilde{P} - \mu, b_j \rangle)$$

for a small enough $\epsilon > 0$.

If the filtering coefficients $(c_j)_{1\le j\le\kappa^2}$ also depend on $\tilde{P}$, like those in (1.2), SURE's expression generally becomes rather unwieldy. In this case, we treat them as constants as an expedient approximation.

3.4.1. Performance Measurement of Adaptive Filters. A useful statistic, the *SURE empirical mean*, can be constructed to measure how effective filters are at denoising. Note that in a conventional filtering scheme, neighboring patches are allowed to overlap one another to help reduce artifacts in restored images. Hence our i.i.d. assumption does not apply (though it does not prevent us from using EM for inference). However, given their restricted supports, it is plausible that patches in a natural image, seen as a two-dimensional stochastic process, satisfy the wide-sense stationarity [2], a weaker condition required to prove the next corollary.

Corollary 3.3. *Under the assumptions of Theorem 3.2 and some mild stationary conditions on image patches* $(P_i)_{1\le i\le N}$ *(Appendix E), the SURE empirical mean*

$$\frac{1}{N}\sum_{i=1}^{N} SURE_f(\tilde{P}_i) := \frac{1}{N}\sum_{i=1}^{N}\Big(\frac{1}{\kappa^2}\|\tilde{P}_i - f(\tilde{P}_i)\|^2 - \sigma^2 + \frac{2\sigma^2}{\kappa^2}\nabla\cdot f(\tilde{P}_i)\Big)$$

is an unbiased estimator of the expected patch MSE $\kappa^{-2}\mathbb{E}[\|P - f(\tilde{P})\|^2]$ *and it converges*

$$\lim_{N\to\infty}\frac{1}{N}\sum_{i=1}^{N} SURE_f(\tilde{P}_i) = \frac{1}{\kappa^2}\mathbb{E}[\|P - f(\tilde{P})\|^2]$$

almost surely and in $\mathbb{L}^2$.

Proof. Sketch of the ideas. Contrary to some other scientific and engineering fields, samples in image processing are usually in abundant supply and the said estimator can be of quite small variance in spite of the terms in the sum being correlated

$$\begin{aligned}
&\frac{1}{N}\sum_{i=1}^{N}\Big(\frac{1}{\kappa^2}\|\tilde{P}_i - f(\tilde{P}_i)\|^2 - \sigma^2 + \frac{2\sigma^2}{\kappa^2}\nabla\cdot f(\tilde{P}_i)\Big)\\
\approx&\mathbb{E}\Big[\frac{1}{N}\sum_{i=1}^{N}\Big(\frac{1}{\kappa^2}\|\tilde{P}_i - f(\tilde{P}_i)\|^2 - \sigma^2 + \frac{2\sigma^2}{\kappa^2}\nabla\cdot f(\tilde{P}_i)\Big)\Big]\\
=&\mathbb{E}\Big[\frac{1}{N}\sum_{i=1}^{N}\frac{1}{\kappa^2}\|P_i - f(\tilde{P}_i)\|^2\Big] \qquad (3.1)\\
=&\frac{1}{\kappa^2}\mathbb{E}[\|P - f(\tilde{P})\|^2] \qquad (3.2)
\end{aligned}$$

where the approximation holds with high probability as a consequence of image patches being stationary together with some additional condition on their covariance structure (Appendix E). Equality (3.1) holds because for all i, we have

$$\mathbb{E}[\frac{1}{\kappa^2}\|\tilde{P}_i - f(\tilde{P}_i)\|^2 - \sigma^2 + \frac{2\sigma^2}{\kappa^2}\nabla\cdot f(\tilde{P}_i)] = \mathbb{E}[\frac{1}{\kappa^2}\|P_i - f(\tilde{P}_i)\|^2]$$

regardless of the prior distribution of P_i thanks to the conditional unbiasedness of SURE and Fubini's theorem. Finally, equality (3.2) stems from $(\|P_i - f(\tilde{P}_i)\|^2)_{1\le i\le N}$ sharing the same expectation, one of the defining conditions of the assumed stationarity. ∎

Due to its dependence on f, the estimator can be seen as a performance measurement of the adaptive filter. Thus one can terminate S-PLE when this estimator goes up in value

because it signals that the algorithm takes a turn to the worse in terms of the produced filtering bases. More importantly, this device provides us with a criterion for switching among filters. In our context, we can let both Wiener (1.2) and shrinkage filters [9], [8] process the noisy patches and then decide the optimal filter for each mixture component by comparing their respective model-wide SURE empirical mean. Our experiments confirmed that with the hard shrinkage and Wiener filter to choose from, the restored image improves in MSE. Nonetheless, it should be emphasized that this rule is not well founded if applied on a patch-by-patch basis because SURE, after all, is a random variable.

3.4.2. Asymptotic Upper Bound on MSE. Perhaps more interestingly, a link can be established between the SURE empirical mean and the MSE of the recovered image. To carry out the analysis, we are in need of the next convention:

`Convention`: the *coordinates* of a patch are those of the pixel located at its up-left corner. A non-overlapping filtering scheme including (p, q) means a partition of $\mathbb{Z}^2$ consisting of patches of dimension $\kappa \times \kappa$ whose coordinates form the set $\{(x, y), (x - p) \bmod \kappa = 0 \text{ and } (y - q) \bmod \kappa = 0\}$.

It is easy to see that there are in total κ^2 distinct filtering schemes. Assume that all of them are used for denoising a large noisy image $\tilde{U}$ of dimension $M_U \times N_U$ which results in $(\widehat{U}_i)_{1\leq i\leq \kappa^2}$. For simplicity, pixels near the image boundary are left untreated because the patches containing them do not lie completely in the image. Corollary 3.3 implies that for all i the restored image MSE $M_U^{-1}N_U^{-1}\|U - \widehat{U}_i\|^2$ (U denotes the original image) will be close to $\kappa^{-2}\mathbb{E}[\|P - f(\tilde{P})\|^2]$ as those bordering pixels do not count if both N_U and M_U are big enough. Jensen's inequality leads to

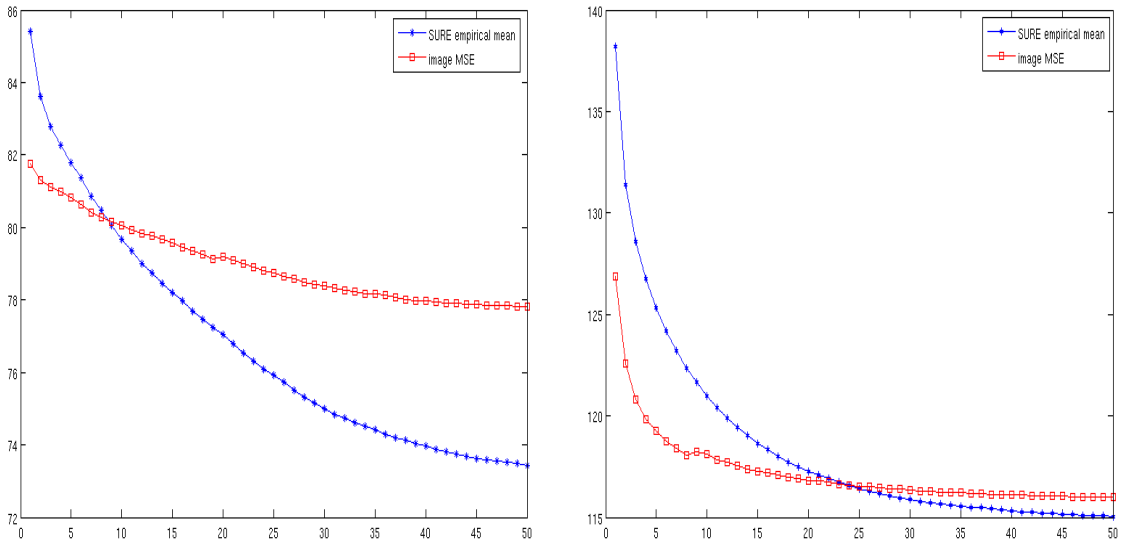
$$\frac{1}{M_U N_U}\|\frac{1}{\kappa^2}\sum_{i=1}^{\kappa^2}\widehat{U}_i - U\|^2 \leq \frac{1}{M_U N_U \kappa^2}\sum_{i=1}^{\kappa^2}\|\widehat{U}_i - U\|^2 \approx \frac{1}{\kappa^2}\mathbb{E}[\|P - f(\tilde{P})\|^2].$$

A similar asymptotic reasoning shows that $M_U^{-1}N_U^{-1}\|\kappa^{-2}\sum_{i=1}^{\kappa^2}\widehat{U}_i - U\|^2$ is close to the MSE of the restored image if a conventional sliding window is used because the majority of pixels in $\kappa^{-2}\sum_{i=1}^{\kappa^2}\widehat{U}_i$ are denoised exactly κ^2 times.

Hence we have one more reason to monitor the SURE empirical mean and let S-PLE run as long as it continues to decrease in value. Although in theory this approach cannot ensure a strict decline of the true MSE, it turned out to be quite reliable in our experiments (Fig. 3.3).

3.4.3. Self-adjusting Flat Patches. The patch size is of importance in the present algorithm. On the one hand, smaller patches lead to a faster algorithm because computational cost of matrix operations usually grows polynomially fast as the patch size increases. Smaller patches also tend to produce less artifacts in the restored image for they introduce less blur. On the other hand, a patch still needs to be large enough to maintain sufficient information for orientation discrimination, though not too big or its content will then become so rich that the orientation ceases to be an adequate concept to characterize the majority of them. Thus, for relatively low noise levels, we settle for a patch size 8×8.

Yet, for noisier images, it is necessary to increase the patch size in order to denoise more aggressively, especially in slow varying areas depicting, for example, sky or building facades.



(a) (b)

Figure 3.3. *(a)* $\sigma = 20$*: the MSE of the restored* `traffic` *and their corresponding SURE empirical mean at each S-PLE iteration (b)* $\sigma = 20$*: the MSE of the restored* `valldemossa` *and their SURE empirical mean at each S-PLE iteration. These two statistics are indeed quite close. The observed deviation from the expected asymptotic behavior could be due to the calculated SURE being biased because of the explained approximation used in dealing with non-linear Wiener filtering coefficients.*

It is hard to overstate the importance of an algorithm's effectiveness at reducing noise in these regions because of the sheer proportion of pixels composing them in natural images.

To help understand the sub-optimality caused by a fixed patch size, consider a slow varying one-dimensional real-valued and time-indexed signal x_τ corrupted by some additive white noise n_τ. For simplicity, let us assume that both the signal and the noise are defined on the integer lattice. If the sliding windows of length $\mathfrak{m}$ are applied to recover the signal, its estimated value at time t is

$$\widehat{x}_t = \frac{1}{\mathfrak{m}} \sum_{\tau=t-\mathfrak{m}+1}^{t} \sum_{s=0}^{\mathfrak{m}-1} \frac{x_{\tau+s} + n_{\tau+s}}{\mathfrak{m}} = \frac{1}{\mathfrak{m}^2} \sum_{s=1-\mathfrak{m}}^{\mathfrak{m}-1} \rho_s (x_{t+s} + n_{t+s}) \tag{3.3}$$

where $\rho_\cdot$ is a probability kernel whose support consists of $2\mathfrak{m} - 1$ elements. Now a simple argument from spectral analysis sheds light on the benefit brought about by an adaptive patch size: when noise level is low, although a bigger patch size is conducive to a higher rate of noise reduction, this gain is not significant enough to counterbalance the loss in signal. However, as noise increases in strength, a small patch size tends to keep more and more noise which ultimately leads to a poor MSE.

To attain a good denoising quality, the basic idea is to seek similar patches so as to enable sparse coding and avoid creating blur. Both block matching and more elaborate 3D collaborative filtering first espoused by NLM [3] and BM3D [6] respectively embody this principle. NLM initially, however, had a somewhat arbitrarily fixed size of smoothing window

for *all* pixels, which prevented NLM from reaching its full potential. Kervrann and Boulanger [14], [15] removed this artificial restriction with a statistical rule for deciding locally optimal window size to minimize MSE.

But in our setting, because of the entirely patch-based framework, it is more natural to view the same problem as that of a patch, rather than window, size selection. Unlike some of the aforementioned algorithms which pay little attention to the meaning of similarity in statistical terms by discounting correlation of noise in overlapping patches, when denoising a patch P, we focus on its non-overlapping neighboring patches which do not intersect with P. One such patch Q is deemed similar to P only if the following two conditions hold simultaneously:

1. both patches belong to the same flat region;
2. the hypothesis that the true states of P and Q are the same shall be upheld statistically;

The first condition can be easily checked thanks to the patch map and the connected component labelling algorithm [25] while the second one simply boils down to a chi-square test: under the null hypothesis, the squared sum of the pixels in $\frac{P-Q}{\sqrt{2}\sigma}$ should follow a chi-square distribution with $\kappa^2 = 64$ degrees, whose law is denoted by $\mathbb{P}_{test}$ in what follows.

Once these supposedly flat and similar patches are identified, they are merged to form a new patch. The fact that they do not overlap amounts to little more than an expansion of the patch P itself. By taking the arithmetic mean of noisy pixels contained in it, one can get a new estimate for the expanded patch. The chi-square test threshold $\mathfrak{t} = 65$ is thus set to verify

$$\mathbb{P}_{test}\big(\frac{\|P-Q\|^2}{2\sigma^2} \leq \mathfrak{t}\big) = 0.5$$

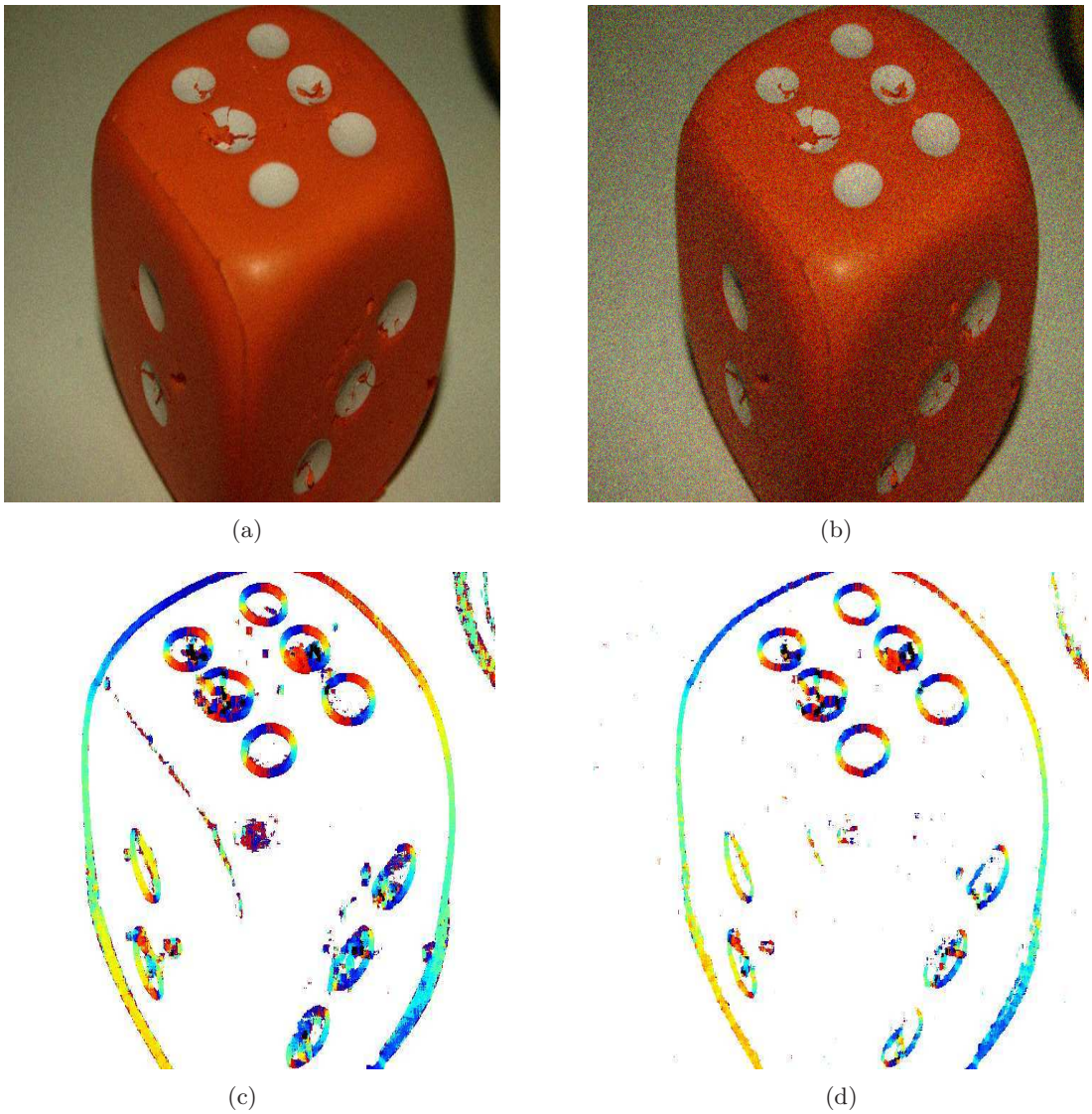
which takes into account the possibly overly simplified null hypothesis and ensures a high likelihood for retaining at least one additional patch to help denoise P.

It ought to be mentioned that when noise is strong, EM can mistake weak borders for noise and cause some patch orientation not properly recognized, which usually happens in the areas of subtle and gradual color transition (Fig.3.4). It is the chi-square test that provides a remedy by favoring the locality of patch blending and thus enhances the algorithm's robustness.

Nonetheless, this adjustment can be problematic when noise level is low. It should be kept in mind that the patch expansion is only justified if it represents a better balance between noise removal and signal preservation. Once again, SURE is the decision aid which we can fall back on: we identify the pixels which only belong to flat patches and put them into a vector $\tilde{\mathfrak{p}}$. Regardless of whether flat patches expand or not, these pixels are restored by simple linear operations similar to those in equation (3.3). The patch size increase can thus be validated or invalidated by comparing the SURE estimates resulting from these two filters operating on $\tilde{\mathfrak{p}}$.

This device enabling automatic patch size selection in relatively flat areas of an image improves dramatically the visual quality as well as the overall MSE of restored images especially when noise is strong.

Finally, we mention that in S-PLE, EM is merely a means to categorize patches and the patch map is heavily relied upon because it hints at the best filter for denoising individual patches. This explains why S-PLE performs less well vis-à-vis other state-of-the-art algorithms



(a) (b)

(c) (d)

Figure 3.4. *The first two images are* `dice` *corrupted by Gaussian noise with (a)* $\sigma = 10$ *and (b)* $\sigma = 30$ *respectively. EM iterated twice on the first transformed channel as explained in Fig.3.2 to produce the patch map for (c)* $\sigma = 10$ *and (d)* $\sigma = 30$*. Notice that the oriented edge on the top side of the dice failed to be recognized at* $\sigma = 30$*.*

Algorithm 4 Flat Patch Expansion

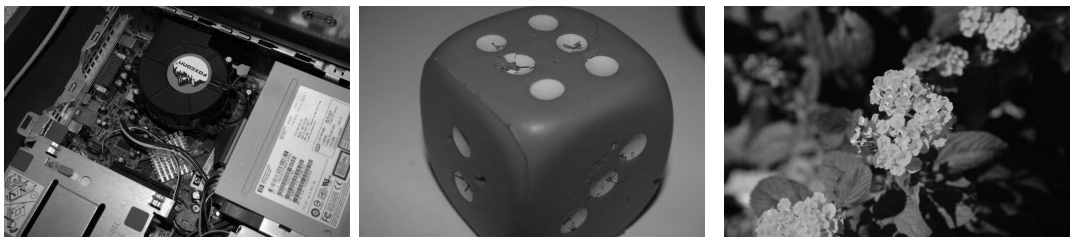
Input: Patch map $\mathcal{M}$, noise level σ and noisy patches $(\tilde{P}_i)_{1\leq i\leq N}$.
Parameters: Patch dimension $\kappa\times\kappa$, search window size $\mathfrak{w}$ and similarity threshold $\mathfrak{t}$.
Run a connected component labeling algorithm on the patch map to locate flat areas.
Identify the pixels which only belong to flat patches and put them into a column vector $\tilde{\mathfrak{p}}\in\mathbb{R}^{\mathfrak{n}_f}$.
for $i=1$ to N **do**
  if $\tilde{P}_i$ belongs to the flat model **then**
    Find, within the search window centered on $\tilde{P}_i$, non-overlapping similar patches sitting in the same flat area as $\tilde{P}_i$.
    Merge them with $\tilde{P}_i$ to have the expanded noisy patch $\tilde{P}_i^e$.
    Estimate all pixels in $\tilde{P}_i^e$ by their arithmetic average which results in $\widehat{P}_i^e$.
    Record in a $\mathfrak{n}_f\times\mathfrak{n}_f$ matrix $\mathfrak{F}_i^e$ the filter used in the previous step so that $\mathfrak{F}_i^e\tilde{\mathfrak{p}}$ and $\widehat{P}_i^e$ coincide on those pixels they share.
  end if
end for
Assign all the filtered patches $\widehat{P}_i^e$ the same weight and restore noisy flat patches. Find the coefficients α_i to have $\mathfrak{F}^e=\sum_i\alpha_i\mathfrak{F}_i$ and $\widehat{\mathfrak{p}}^e=\mathfrak{F}^e\tilde{\mathfrak{p}}$ where $\widehat{\mathfrak{p}}^e$ denotes the restored pixels on the same sites as those in $\tilde{\mathfrak{p}}$.
Calculate the resulting SURE $\mathfrak{S}^e$.
Repeat the same steps without expanding flat patches and denote the SURE estimate $\mathfrak{S}$.
if $\mathfrak{S}^e<\mathfrak{S}$ **then**
  Take the estimates with patch expansion.
else
  Take the estimates without patch expansion.
end if

in presence of strong noise ($\sigma = 40$ for example) because its patch classification becomes less reliable.

4. Algorithm Outline and A Comparative Study. To reduce execution time at denoising a color image, instead of asking the computationally intensive EM algorithm to run on three transformed channels, the first transformed channel, supposedly with the highest SNR, should be given priority so that EM only operates on this channel and can iterate more rounds than otherwise. The other two channels are then restored using the same patch map and filters resulting from these iterations. This expedient solution is backed by the obervation that more iterations on the first transformed channel generally bring about better results in terms of MSE.

It is also observed that, at times, some component's mixing weight gets close to zero after several EM iterations due to its under-representation in sample patches. To accelerate the algorithm and potentially achieve a better data fit, one can choose to eliminate those models that seem to account for nothing but a tiny fraction of patches. In so doing, however, the monotone convergence of the overall data likelihood might no longer be guaranteed. But some experimental data showed that the model culling does not affect in a significant way the denoising algorithm's performance.

Table 4.1 compares S-PLE [30] with several other acclaimed algorithms whose implementations [16], [18], [17], [4], [31], [22] are available on the `IPOL` website. Since noise is random, what we really wish to compare is the mean RMSEs various algorithms can achieve given the same noiseless image. But as an algorithm operating on a big image usually produces a quite stable RMSE (whose empirical standard deviation rarely exceeds 0.05), we thus feed independently generated noisy images to each algorithm just once before compiling the results.



(a) (b) (c)

(d) (e) (f)

Figure 4.1. *gray images used in algorithm comparison (a) computer* (704 × 469) *(b) dice* (704 × 469) *(c) flowers* (704 × 469) *(d) girl* (704 × 469) *(e) traffic* (704 × 469) *(f) valldemossa* (769 × 338)

Algorithm 5 S-PLE

Input: A noisy gray image $\tilde{U}$.
Parameter: Number of EM iteration S, noise level σ.
Read in the GMM setup Θ_0 and set the initial SURE empirical mean to $\mathfrak{E}_0 = (\sigma+1)^2$ to reflect its interpretation as an asymptotic MSE upper bound. Extract all 8×8 patches from $\tilde{U}$ to form the noisy patch set $\mathcal{P}$ and compute their posterior probabilities $\forall \tilde{P} \in \mathcal{P},\ \forall 0 \le k \le 19,\ \mathbb{P}_{\Theta_0}(s_P = k \mid \tilde{P})$.

for $t = 1$ to S **do**
Update model priors:

$$\forall 0 \le k \le 19,\ \ \boldsymbol{w}_{k,t} = \frac{1}{|\mathcal{P}|} \sum_{\tilde{P} \in \mathcal{P}} \mathbb{P}_{\Theta_{t-1}}(s_P = k \mid \tilde{P}).$$

Update model means:

$$\forall 0 \le k \le 19,\ \ \boldsymbol{\mu}_{k,t} = \frac{\sum_{\tilde{P} \in \mathcal{P}} \tilde{P} \mathbb{P}_{\Theta_{t-1}}(s_P = k \mid \tilde{P})}{\sum_{\tilde{P} \in \mathcal{P}} \mathbb{P}_{\Theta_{t-1}}(s_P = k \mid \tilde{P})}.$$

Update factor loadings:

$$\forall 0 \le k \le 19,\ \ \boldsymbol{F}_{k,t} = \tilde{\boldsymbol{\Sigma}}^*_{k,t-1} \boldsymbol{F}_{k,t-1} (M^{-1}_{k,t-1} \boldsymbol{F}^T_{k,t-1} \tilde{\boldsymbol{\Sigma}}^*_{k,t-1} \boldsymbol{F}_{k,t-1} + \sigma^2 I_{l_k})^{-1}.$$

with $l_k = 32$ for all k except for the last two: $l_{18} = 63$ and $l_{19} = 1$ where

$$\forall 0 \le k \le 19,\ \ M_{k,t-1} = \boldsymbol{F}^T_{k,t-1} \boldsymbol{F}_{k,t-1} + \sigma^2 I_{l_k} \text{ and } \tilde{\boldsymbol{\Sigma}}^*_{k,t-1} = \frac{\sum_{\tilde{P} \in \mathcal{P}} (\tilde{P} - \boldsymbol{\mu}_{k,t})(\tilde{P} - \boldsymbol{\mu}_{k,t})^T \mathbb{P}_{\Theta_{t-1}}(s_P = k \mid \tilde{P})}{\sum_{\tilde{P} \in \mathcal{P}} \mathbb{P}_{\Theta_{t-1}}(s_P = k \mid \tilde{P})}.$$

For all k, apply the spectral decomposition to $\boldsymbol{F}_{k,t} \boldsymbol{F}^T_{k,t}$ to have its l_k orthonormal leading eigenvectors.
Create the patch map with the updated parameter set Θ_t:

$$\mathcal{M} : \tilde{P} \in \mathcal{P} \mapsto \operatorname*{argmax}_{0 \le k \le 19} \mathbb{P}_{\Theta_t}(s_P = k \mid \tilde{P}).$$

For all k, denoise the patches assigned to the k-th model with both Wiener and the hard shrinkage filter and pick the better filtered patches according to their achieved model-wide SURE empirical mean.
Record the SURE empirical mean $\mathfrak{E}_t$.
Try patch expansion in flat areas.
if $\mathfrak{E}_t > \mathfrak{E}_{t-1}$ **then**
Break (Or continue iterating to see if the SURE empirical mean will eventually go below $\mathfrak{E}_{t-1}$).
end if
end for
Assign equal weights to all restored patches and recover the image.

Table 4.1
Algorithm Comparison[1]

[3]$\sigma = 2$	[5]PLE	DCT	[4]GSM	KSVD	NLM	EPLL	[2]S-PLE	BM3D	NLBayes
computer	2.40	1.65	1.64	1.55	1.64	1.57	1.54	**1.52**	1.85
dice	0.96	0.91	0.92	0.96	0.97	0.89	0.86	**0.84**	1.31
flowers	1.25	1.08	1.09	1.09	1.29	1.09	**1.02**	1.04	1.44
girl	1.24	1.13	1.12	1.14	1.17	1.09	1.09	**1.05**	1.50
traffic	2.82	1.73	1.77	1.65	1.72	1.64	1.67	**1.62**	1.97
valldemossa	3.65	1.75	1.79	1.73	1.76	1.69	1.78	**1.68**	2.12
average	2.05	1.37	1.38	1.35	1.42	1.32	1.32	**1.29**	1.69

$\sigma = 5$	PLE	DCT	GSM	KSVD	NLM	EPLL	S-PLE	BM3D	NLBayes
computer	4.25	3.40	3.28	3.08	3.19	3.05	2.97	**2.94**	2.95
dice	1.45	1.44	1.51	1.89	1.70	1.32	1.29	**1.27**	1.72
flowers	2.16	1.97	1.97	2.11	2.42	1.87	**1.79**	1.81	2.18
girl	1.92	1.85	1.89	2.11	2.01	1.74	**1.69**	**1.69**	1.93
traffic	4.84	3.76	3.69	3.49	3.70	**3.38**	**3.38**	3.40	3.63
valldemossa	6.48	4.04	3.98	3.90	4.15	**3.75**	3.81	3.77	3.85
average	3.51	2.74	2.72	2.76	2.86	2.51	**2.48**	**2.48**	2.71

$\sigma = 10$	PLE	DCT	GSM	KSVD	NLM	EPLL	S-PLE	BM3D	NLBayes
computer	6.12	5.66	5.36	5.14	5.16	4.89	4.77	4.65	**4.51**
dice	2.08	2.08	2.24	3.42	2.80	1.90	**1.80**	1.82	2.15
flowers	3.26	3.14	3.19	3.70	4.01	2.92	**2.85**	2.86	3.07
girl	2.65	2.61	2.82	3.60	3.21	2.44	**2.35**	**2.35**	2.56
traffic	7.18	6.51	6.21	5.99	6.05	5.61	5.68	5.67	**5.57**
valldemossa	9.24	7.45	7.04	6.94	7.02	6.58	6.65	6.66	**6.51**
average	5.08	4.57	4.47	4.79	4.70	4.05	**4.00**	**4.00**	4.06

$\sigma = 20$	PLE	DCT	GSM	KSVD	NLM	EPLL	S-PLE	BM3D	NLBayes
computer	8.86	8.82	8.37	8.56	7.90	7.54	7.41	7.18	**7.07**
dice	3.20	3.05	3.19	6.74	3.55	2.95	**2.66**	2.67	2.76
flowers	4.97	4.88	5.02	6.60	5.66	4.57	4.55	**4.48**	4.67
girl	3.84	3.65	4.33	6.55	4.18	3.55	3.35	**3.28**	3.40
traffic	10.37	10.08	9.82	9.71	9.40	**8.70**	8.80	8.83	8.74
valldemossa	13.26	12.26	11.55	11.47	11.19	10.60	10.73	10.77	**10.53**
average	7.41	7.12	7.04	8.27	6.98	6.31	6.25	6.20	**6.19**

$\sigma = 30$	PLE	DCT	GSM	KSVD	NLM	EPLL	S-PLE	BM3D	NLBayes
computer	10.97	11.13	10.83	10.22	10.43	9.51	9.39	**9.09**	9.12
dice	4.43	3.88	4.18	6.09	4.87	3.94	3.37	3.44	**3.35**
flowers	6.44	6.37	6.15	6.95	7.45	6.00	5.92	**5.80**	5.89
girl	4.85	4.46	4.64	6.24	5.45	4.51	4.11	**4.04**	4.10
traffic	12.23	12.38	12.35	11.58	12.11	**10.85**	11.08	10.97	10.99
valldemossa	15.80	15.32	14.74	14.20	14.37	**13.33**	13.58	13.64	13.43
average	9.12	8.92	8.81	9.21	9.11	8.02	7.90	7.83	**7.81**

$\sigma = 40$	PLE	DCT	GSM	KSVD	NLM	EPLL	S-PLE	BM3D	NLBayes
computer	12.61	12.92	12.85	12.20	12.41	11.13	11.24	**10.72**	10.85
dice	5.85	4.64	4.96	7.91	5.20	4.80	4.49	4.14	**3.95**
flowers	7.68	7.59	7.32	8.55	8.96	7.12	7.14	**6.94**	6.98
girl	6.07	5.23	6.01	7.86	5.84	5.30	5.17	4.67	**4.60**
traffic	13.87	14.17	14.70	13.61	14.24	**12.53**	12.86	12.70	12.90
valldemossa	17.71	17.48	17.22	16.52	16.90	**15.57**	15.83	15.73	15.62
average	10.63	10.33	10.51	11.10	10.59	9.40	9.45	**9.15**	**9.15**

[1] the algorithms are ordered to reflect their global performance. In bold is the lowest RMSE in each row.
[2] S-PLE was allowed to iterate 50 times.
[3] noise standard deviation
[4] BLS-GSM [21]
[5] PLE, with no observable convergence available, iterated four times

Figure 4.2. *(a) original image (b) noisy image* $\sigma = 20$ *(c) EPLL RMSE* $= 3.55$ *(d) S-PLE RMSE* $= 3.35$ *(e) BM3D RMSE* $= 3.28$ *(f) NLBayes RMSE* $= 3.40$

5. How About A Richer Prior?. In view of the comparison table, it is clear that S-PLE presents a huge improvement over the original PLE and achieves a level of denoising performance worthy of being counted as another state-of-the-art algorithm. These best algorithms' unmistakable convergence towards a similar end result in RMSE is striking indeed, a phenomenon which has already been taken note of and spurred an interesting discussion [20], [19] of whether the academic endeavour in this field of research has effectively come to an end. In

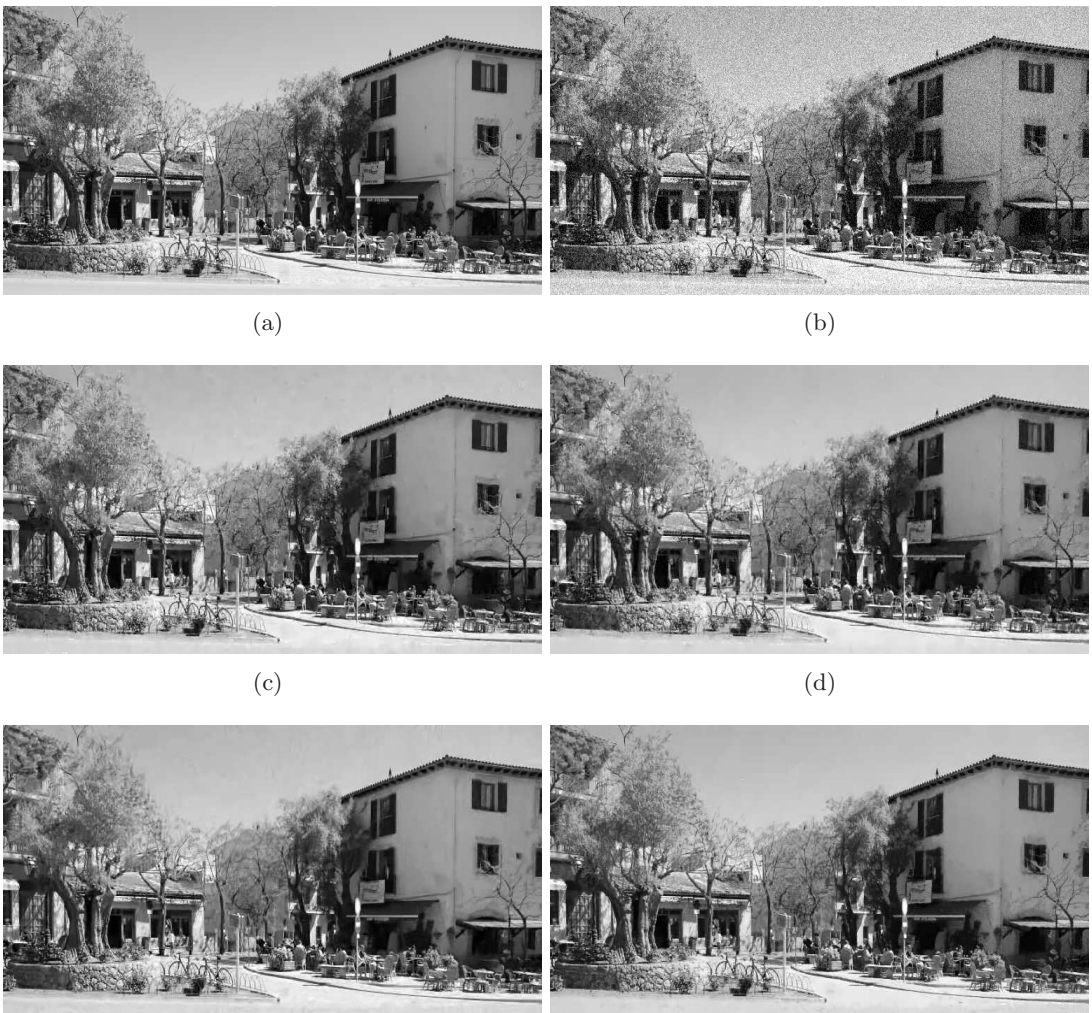
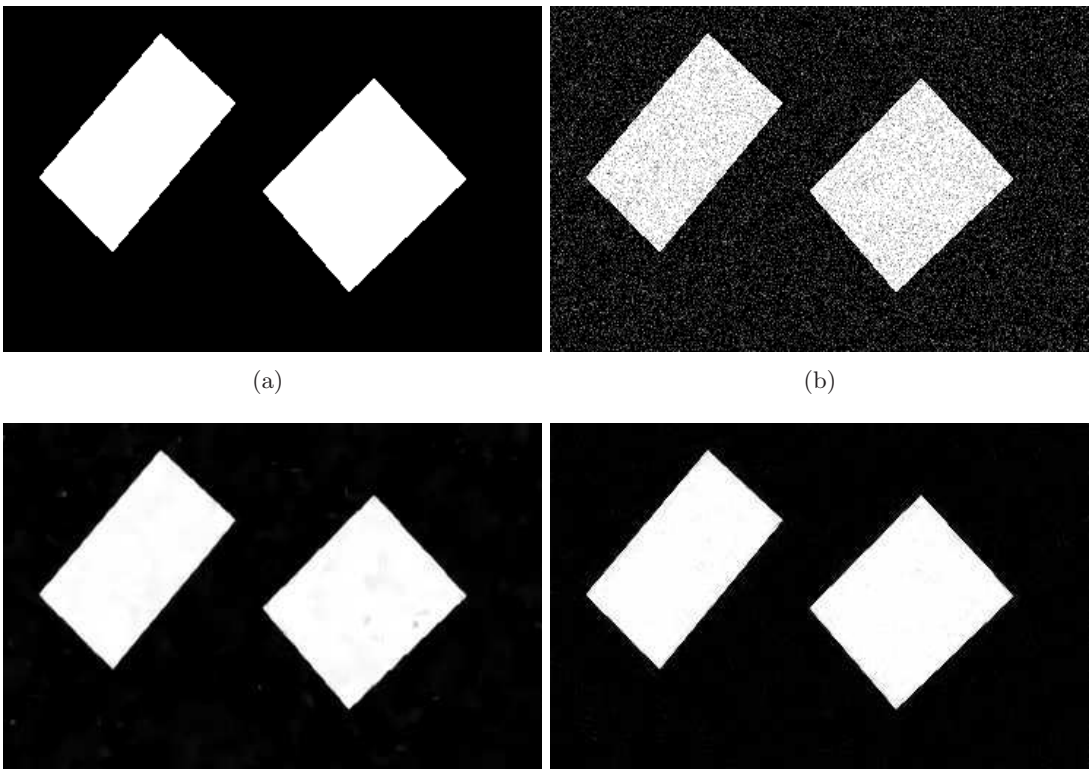


Figure 4.3. *(a) original image (b) noisy image* $\sigma = 20$ *(c) EPLL RMSE* $= 10.60$ *(d) S-PLE RMSE* $= 10.73$ *(e) BM3D RMSE* $= 10.77$ *(f) NLBayes RMSE* $= 10.53$

addition, it is also surprising to see that S-PLE, in possession of only 20 models, is capable of yielding competitive result vis-à-vis its counterparts which make use of at least 10 times more Gaussian models, produced either locally (NLM and its variants) or by extensive preliminary sampling (EPLL). In our opinion, this significant model number reduction is made possible as a result of the initial models' representativeness as well as the algorithm's ability to adjust and reuse them in the process of overall likelihood maximization. To stave off potential confusion, let it be clear that, compared with S-PLE, EPLL has a fundamentally different and more appealing data fidelity term which allows direct likelihood evaluation of individual patches as they appear in a restored image. Yet, the question we attempt to answer is whether local image learning can make up for a possibly less suitable prior.



(a) (b)

(c) (d)

Figure 5.1. *Local learning is a remedy against improper initial prior: (a) clean image (b) noisy image ($\sigma = 50$) (c) EPLL restored RMSE $= 8.15$ (d) S-PLE restored RMSE $= 5.16$ (20 iterations). As EPLL draws its prior from natural images, its restoration suffers from artifacts, an issue S-PLE shares at the beginning but largely manages to address thanks to learning.*

We ran S-PLE with all models except for the flat one replaced by those used in EPLL [33]. No factor trimming was performed for these 200 Gaussian components who jointly accounted for an initial prior probability equal to 0.6 with their relative importance kept as they are in EPLL, leaving the flat model responsible for the rest forty percent of patches. Conversely we also let EPLL operate using our 20 initial models. The results effectively support our claim that unless the Gaussian mixture prior is kept fixed, 20 well-chosen components appear to be sufficient for the denoising purpose.

Table 5.1

EPLL's denoising performance under two different priors[1]

$\sigma = 2$	computer	dice	flowers	girl	traffic	valldemossa
$\mathbf{EPLL}_{20}$	1.83	1.72	1.74	1.73	1.86	1.90
$\mathbf{EPLL}_{200}$	1.57	0.89	1.09	**1.09**	**1.64**	**1.69**
$\mathbf{S\text{-}PLE}_{20}^{50}$	**1.54**	**0.86**	**1.02**	**1.09**	1.67	1.78

$\sigma = 5$	computer	dice	flowers	girl	traffic	valldemossa
$\mathbf{EPLL}_{20}$	3.65	3.00	3.10	3.08	3.87	4.15
$\mathbf{EPLL}_{200}$	3.05	1.32	1.87	1.74	**3.38**	**3.75**
$\mathbf{S\text{-}PLE}_{20}^{50}$	**2.97**	**1.29**	**1.79**	**1.69**	**3.38**	3.81

$\sigma = 10$	computer	dice	flowers	girl	traffic	valldemossa
$\mathbf{EPLL}_{20}$	5.62	3.89	4.22	4.03	6.22	7.09
$\mathbf{EPLL}_{200}$	4.89	1.90	2.92	2.44	**5.61**	**6.58**
$\mathbf{S\text{-}PLE}_{20}^{50}$	**4.77**	**1.80**	**2.85**	**2.35**	5.68	6.65

$\sigma = 20$	computer	dice	flowers	girl	traffic	valldemossa
$\mathbf{EPLL}_{20}$	8.19	4.71	5.60	4.97	9.30	11.19
$\mathbf{EPLL}_{200}$	7.54	2.95	4.57	3.55	**8.70**	**10.60**
$\mathbf{S\text{-}PLE}_{20}^{50}$	**7.41**	**2.66**	**4.55**	**3.35**	8.80	10.73

$\sigma = 30$	computer	dice	flowers	girl	traffic	valldemossa
$\mathbf{EPLL}_{20}$	10.07	5.33	6.77	5.65	11.36	13.93
$\mathbf{EPLL}_{200}$	9.51	3.94	6.00	4.51	**10.85**	**13.33**
$\mathbf{S\text{-}PLE}_{20}^{50}$	**9.39**	**3.37**	**5.92**	**4.11**	11.08	13.58

$\sigma = 40$	computer	dice	flowers	girl	traffic	valldemossa
$\mathbf{EPLL}_{20}$	11.61	5.96	7.71	6.22	13.00	16.10
$\mathbf{EPLL}_{200}$	**11.13**	4.80	**7.12**	5.30	**12.53**	**15.57**
$\mathbf{S\text{-}PLE}_{20}^{50}$	11.24	**4.49**	7.14	**5.17**	12.86	15.83

[1] $\mathbf{EPLL}_{20}$ and $\mathbf{EPLL}_{200}$ refer to EPLL run with a Gaussian mixture composed of 20 and 200 components respectively. $\mathbf{S\text{-}PLE}_{20}^{50}$ 50 iterations of S-PLE with 20 components. Algorithms are ordered to reflect their global performance. Marked in bold is the lowest RMSE in each column.

Table 5.2

S-PLE's denoising performance under two different priors[1]

$\sigma = 2$	computer	dice	flowers	girl	traffic	valldemossa
$\mathbf{S\text{-}PLE}_{20}^{10}$	1.58	0.88	1.04	1.12	1.69	1.80
$\mathbf{S\text{-}PLE}_{201}^{10}$	1.57	0.87	1.03	**1.09**	**1.64**	**1.69**
$\mathbf{S\text{-}PLE}_{20}^{50}$	**1.54**	**0.86**	**1.02**	**1.09**	1.67	1.78

$\sigma = 5$	computer	dice	flowers	girl	traffic	valldemossa
$\mathbf{S\text{-}PLE}_{20}^{10}$	3.03	1.31	1.82	1.73	3.43	3.85
$\mathbf{S\text{-}PLE}_{201}^{10}$	2.99	1.31	1.85	1.74	**3.38**	**3.78**
$\mathbf{S\text{-}PLE}_{20}^{50}$	**2.97**	**1.29**	**1.79**	**1.69**	**3.38**	3.81

$\sigma = 10$	computer	dice	flowers	girl	traffic	valldemossa
$\mathbf{S\text{-}PLE}_{20}^{10}$	4.85	1.82	2.90	2.38	5.75	6.71
$\mathbf{S\text{-}PLE}_{201}^{10}$	4.78	1.87	2.94	2.46	**5.64**	**6.63**
$\mathbf{S\text{-}PLE}_{20}^{50}$	**4.77**	**1.80**	**2.85**	**2.35**	5.68	6.65

$\sigma = 20$	computer	dice	flowers	girl	traffic	valldemossa
$\mathbf{S\text{-}PLE}_{20}^{10}$	7.51	**2.66**	4.65	3.36	8.93	10.83
$\mathbf{S\text{-}PLE}_{201}^{10}$	**7.39**	2.79	4.75	3.57	8.82	**10.72**
$\mathbf{S\text{-}PLE}_{20}^{50}$	7.41	**2.66**	**4.55**	**3.35**	**8.80**	10.73

$\sigma = 30$	computer	dice	flowers	girl	traffic	valldemossa
$\mathbf{S\text{-}PLE}_{20}^{10}$	9.52	3.57	6.06	4.28	11.22	13.72
$\mathbf{S\text{-}PLE}_{201}^{10}$	9.47	3.62	6.17	4.38	**11.07**	**13.58**
$\mathbf{S\text{-}PLE}_{20}^{50}$	**9.39**	**3.37**	**5.92**	**4.11**	11.08	**13.58**

$\sigma = 40$	computer	dice	flowers	girl	traffic	valldemossa
$\mathbf{S\text{-}PLE}_{20}^{10}$	11.28	4.49	7.74	5.17	13.02	15.92
$\mathbf{S\text{-}PLE}_{201}^{10}$	**11.13**	**4.35**	7.48	**5.07**	**12.82**	**15.80**
$\mathbf{S\text{-}PLE}_{20}^{50}$	11.24	4.49	**7.14**	5.17	12.86	15.83

[1] $\mathbf{S\text{-}PLE}_{\mathbf{c}}^{\mathbf{t}}$ refers to S-PLE run **t** iterations with a Gaussian mixture composed of **c** components. Algorithms are ordered to reflect their global performance. Marked in bold is the lowest RMSE in each column.

Appendix A. Optimal Global Linear Filter for GMM.

To solve the optimization problem (1.1)

$$\mathbb{E}[\|\boldsymbol{L}\tilde{P}+\boldsymbol{b}-P\|_2^2] \tag{A.1}$$

$$=\mathbb{E}[\|\sum_{k=0}^{K-1}1_{s_P=k}(\boldsymbol{L}P+\boldsymbol{L}N+\boldsymbol{b}-P)\|_2^2]$$

$$=\sum_{k=0}^{K-1}w_k\Big(\|(\boldsymbol{L}-I)\mu_k+\boldsymbol{b}\|_2^2+tr(\sigma^2\boldsymbol{L}\boldsymbol{L}^T+(\boldsymbol{L}-I)\Sigma_k(\boldsymbol{L}-I)^T)\Big), \tag{A.2}$$

we proceed in two steps: first, if the optimal $\boldsymbol{L}_*$ is found, $\boldsymbol{b}_*$ can be determined by minimizing the quadratic function (A.2)

$$\frac{d}{d\boldsymbol{b}}\sum_{k=0}^{K-1}w_k\|(\boldsymbol{L}_*-I)\mu_k+\boldsymbol{b}\|_2^2=0\Rightarrow\boldsymbol{b}_*=-(\boldsymbol{L}_*-I)\sum_{k=0}^{K-1}w_k\mu_k.$$

Second, by substituting $(\boldsymbol{L}_*,\boldsymbol{b}_*)$ for $(\boldsymbol{L},\boldsymbol{b})$ in the expectation (A.1), we find

$$\sum_{k=0}^{K-1}w_k\|(\boldsymbol{L}_*-I)(\mu_k-\sum_{i=0}^{K-1}w_i\mu_i)\|_2^2+tr\Big(\sigma^2\boldsymbol{L}_*\boldsymbol{L}_*^T+(\boldsymbol{L}_*-I)(\sum_{k=0}^{K-1}w_k\Sigma_k)(\boldsymbol{L}_*-I)^T\Big)$$

$$=tr\Big(\sigma^2\boldsymbol{L}_*\boldsymbol{L}_*^T+(\boldsymbol{L}_*-I)\boldsymbol{C}(\boldsymbol{L}_*-I)^T\Big)$$

where $\boldsymbol{C}$ denotes the blended covariance matrix

$$\boldsymbol{C}=\sum_{k=0}^{K-1}w_k(\Sigma_k+\Omega_k)\quad\text{with}\quad\Omega_k:=(\mu_k-\sum_{i=0}^{K-1}w_i\mu_i)(\mu_k-\sum_{i=0}^{K-1}w_i\mu_i)^T.$$

It follows

$$\boldsymbol{L}_*=\boldsymbol{C}(\boldsymbol{C}+\sigma^2I)^{-1}$$

which is the conventional signal-to-noise (SNR) interpretation of Wiener filtering. To conclude, the best linear estimator of a noisy patch $\tilde{P}$ is

$$\widehat{P}=\boldsymbol{L}_*\tilde{P}+\boldsymbol{b}_*=\boldsymbol{L}_*(\tilde{P}-\sum_{k=0}^{K-1}w_k\mu_k)+\sum_{k=0}^{K-1}w_k\mu_k.$$

It is only optimal among the linear filters under the assumption made on the joint distribution of signal and noise, hence the introduction of a host of non-linear denoising schemes.

Appendix B. Conditional Expectation $\mathbb{E}[P|\tilde{P}]$.

Let us state the result in the form of a theorem

Theorem B.1. *Using the paper's notations and hypotheses, we have*

$$\mathbb{E}[P|\tilde{P}]=\sum_{k=0}^{K-1}\mathbb{P}(s_P=k|\tilde{P})\mathbb{E}[P|s_P=k,\tilde{P}].$$

Proof. First observe that the expression on the right hand side of the equation, seen as a random variable, is measurable with respect to the σ-algebra generated by $\tilde{P}$. Thus, we only need to verify that the next equality holds for every element A in the Borelian set $\mathcal{B}(\mathbb{R}^{\kappa^2})$

$$\mathbb{E}[P1_{\tilde{P}\in A}] = \mathbb{E}[\sum_{k=0}^{K-1} \mathbb{P}(s_P = k|\tilde{P})\mathbb{E}[P|s_P = k, \tilde{P}]1_{\tilde{P}\in A}]$$

which is true because

$$\begin{aligned}\sum_{k=0}^{K-1} \mathbb{E}[P1_{s_P=k}1_{\tilde{P}\in A}] &= \sum_{k=0}^{K-1} \mathbb{E}[\mathbb{E}[P|s_P, \tilde{P}]1_{s_P=k}1_{\tilde{P}\in A}] \\ &= \sum_{k=0}^{K-1} \mathbb{E}[\mathbb{E}[P|s_P = k, \tilde{P}]1_{s_P=k}1_{\tilde{P}\in A}] \\ &= \sum_{k=0}^{K-1} \mathbb{E}[\mathbb{E}[P|s_P = k, \tilde{P}]\mathbb{P}(s_P = k|\tilde{P})1_{\tilde{P}\in A}].\end{aligned}$$

The k-indexed functions

$$f_k : \tilde{P} \mapsto \mathbb{E}[P|s_P = k, \tilde{P}],\ k \in \mathbb{Z} \cap [0, K-1]$$

amount to little more than a linear filter because the condition $s_P = k$ effectively restricts P as well as $\tilde{P}$ to behave like an ordinary Gaussian vector, and they read

$$\forall k \in \mathbb{Z} \cap [0, K-1],\ f_k(\tilde{P}) = \boldsymbol{F}_k(\sigma^2 I + \boldsymbol{F}_k^T\boldsymbol{F}_k)^{-1}\boldsymbol{F}_k^T(\tilde{P} - \boldsymbol{\mu}_k) + \boldsymbol{\mu}_k.$$

■

Appendix C. Two-Stage EM Algorithm for Gaussian Factor Mixture.

With the parameter set and the noisy patches denoted by $\Theta := \{\boldsymbol{\sigma}, (\boldsymbol{w}_k, \boldsymbol{\mu}_k, \boldsymbol{F}_k)_{0\le k\le K-1}\}$ and $X := (\tilde{P}_i)_{1\le i\le N}$ respectively, we would like to find the maximum-likelihood estimator (MLE)

$$\Theta^* = \operatorname*{argmax}_{\Theta} p_\Theta(X)$$

which is by itself a hard problem due to the potentially complicated form of the function $\Theta \mapsto p_\Theta(X)$. Expectation-Maximization (EM) completes X with the latent dataset $Y := (s_{P_i})_{1\le i\le N}$, interpreted as the labels recording the generating model of each patch, and introduces an auxiliary function

$$Q_{\Theta_t}(\Theta) = \mathbb{E}_{\Theta_t}[\log p_\Theta(X, Y)|X]$$

with Θ_t representing the estimate of Θ at step t. This conditional expectation is easy to compute thanks to the discreteness of the patch model selector s_P. And we are interested in

the scenario where the completed dataset (X, Y) is made up of i.i.d. samples $(\tilde{P}_i, s_{P_i})_{1\le i\le N}$

$$\begin{aligned}
\mathbb{E}_{\Theta_t}[\,\log p_\Theta(X,Y)|X\,] &= \sum_{i=1}^{N} \mathbb{E}_{\Theta_t}[\,\log p_\Theta(\tilde{P}_i, s_{P_i})|X\,] \\
&= \sum_{i=1}^{N} \mathbb{E}_{\Theta_t}[\,\log p_\Theta(\tilde{P}_i, s_{P_i})|\tilde{P}_i\,] && \text{(C.1)} \\
&= \sum_{i=1}^{N}\sum_{k=0}^{K-1} \mathbb{E}_{\Theta_t}[\,\log p_\Theta(\tilde{P}_i, k)1_{s_{P_i}=k}|\tilde{P}_i\,] \\
&= \sum_{i=1}^{N}\sum_{k=0}^{K-1} \log p_\Theta(\tilde{P}_i, k)\mathbb{P}_{\Theta_t}(s_{P_i}=k\mid\tilde{P}_i) && \text{(C.2)} \\
&= \sum_{i=1}^{N}\sum_{k=0}^{K-1} \log p_\Theta(\tilde{P}_i|s_{P_i}=k)\mathbb{P}_{\Theta_t}(s_{P_i}=k\mid\tilde{P}_i) && \text{(C.3)} \\
&\quad + \sum_{i=1}^{N}\sum_{k=0}^{K-1} \log \mathbb{P}_\Theta(s_{P_i}=k)\mathbb{P}_{\Theta_t}(s_{P_i}=k\mid\tilde{P}_i). && \text{(C.4)}
\end{aligned}$$

Remarks:

(a) all the equalities hold almost everywhere (a.e.);

(b) equality (C.1) follows from the i.i.d. assumption;

(c) equality (C.2) holds because the random variable $\log p_\Theta(\tilde{P}_i, k)$ is measurable with respect to the σ-algebra generated by $\tilde{P}_i$ (or equivalently, the function $\log p_\Theta(\cdot, k)$ is Borelian).

Maximizing $Q_{\Theta_t}(\cdot)$ yields a new set of parameters serving as the next iteration's input: Θ_{t+1}. To carry out the maximization, note that the i.i.d. assumption means that $\boldsymbol{w}_k = \mathbb{P}_\Theta(s_{P_i}=k)$ only depends on k and Θ. Optimizing the second term (C.4) boils down to

$$\begin{aligned}
&\max_{\boldsymbol{w}_0,\cdots,\boldsymbol{w}_{K-1}} \sum_{k=0}^{K-1} \log \boldsymbol{w}_k \sum_{i=1}^{N} \mathbb{P}_{\Theta_t}(s_{P_i}=k\mid\tilde{P}_i) \\
&\text{subject to } \sum_{k=0}^{K-1} \boldsymbol{w}_k = 1,\ \min_{0\le k\le K-1} \boldsymbol{w}_k \ge 0
\end{aligned}$$

and the strong duality holds (Slater condition): the solution to its dual problem is

$$\boldsymbol{w}_{k,t+1} = \frac{\sum_{i=1}^{N} \mathbb{P}_{\Theta_t}(s_{P_i}=k\mid\tilde{P}_i)}{\sum_{k=0}^{K-1}\sum_{i=1}^{N} \mathbb{P}_{\Theta_t}(s_{P_i}=k\mid\tilde{P}_i)} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{P}_{\Theta_t}(s_{P_i}=k\mid\tilde{P}_i).$$

Now turn to the first term (C.3) and seek its maximum

$$\begin{aligned}
&\sum_{i=1}^{N}\sum_{k=0}^{K-1}\log p_\Theta(\tilde{P}_i \mid s_{P_i}=k)\mathbb{P}_{\Theta_t}(s_{P_i}=k\mid \tilde{P}_i)\\
=&\sum_{i=1}^{N}\sum_{k=0}^{K-1}\log\frac{e^{-\frac{1}{2}(\tilde{P}_i-\boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}(\tilde{P}_i-\boldsymbol{\mu}_k)}}{\sqrt{(2\pi)^{\kappa^2}\det(\boldsymbol{\Sigma}_k)}}\mathbb{P}_{\Theta_t}(s_{P_i}=k\mid \tilde{P}_i)\\
=&-\frac{1}{2}\sum_{i=1}^{N}\sum_{k=0}^{K-1}\Big((\tilde{P}_i-\boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}(\tilde{P}_i-\boldsymbol{\mu}_k)+\log(2\pi)^{\kappa^2}+\log\det\boldsymbol{\Sigma}_k\Big)\mathbb{P}_{\Theta_t}(s_{P_i}=k\mid \tilde{P}_i)
\end{aligned}$$

where

$$\forall 0\le k\le K-1,\quad \boldsymbol{\Sigma}_k:=\boldsymbol{F}_k\boldsymbol{F}_k^T+\boldsymbol{\sigma}^2 I_{\kappa^2}.$$

After some trivial simplification, we get a regularization problem

$$\begin{aligned}
&\min_{(\boldsymbol{\mu}_k,\boldsymbol{F}_k)_{0\le k\le K-1},\boldsymbol{\sigma}}\ \sum_{k=0}^{K-1}\Big(tr(\boldsymbol{\Sigma}_k^{-1}\tilde{\boldsymbol{\Sigma}}_{k,t})+\log\det\boldsymbol{\Sigma}_k\Big)\\
&\text{subject to}\quad \boldsymbol{\Sigma}_k=\boldsymbol{F}_k\boldsymbol{F}_k^T+\boldsymbol{\sigma}^2 I_{\kappa^2},\ 0\le k\le K-1\\
&\qquad\qquad \tilde{\boldsymbol{\Sigma}}_{k,t}=\frac{\sum_{i=1}^{N}(\tilde{P}_i-\boldsymbol{\mu}_k)(\tilde{P}_i-\boldsymbol{\mu}_k)^T\mathbb{P}_{\Theta_t}(s_{P_i}=k\mid\tilde{P}_i)}{\sum_{i=1}^{N}\mathbb{P}_{\Theta_t}(s_{P_i}=k\mid\tilde{P}_i)},\ 0\le k\le K-1\\
&\qquad\qquad \boldsymbol{\sigma}\in\mathbb{R}_+.
\end{aligned}$$

Differentiate with respect to $\boldsymbol{\mu}_k$:

$$\frac{d}{d\boldsymbol{\mu}_k}tr(\boldsymbol{\Sigma}_k^{-1}\tilde{\boldsymbol{\Sigma}}_{k,t})=\frac{2\boldsymbol{\Sigma}_k^{-1}\sum_{i=1}^{N}\mathbb{P}_{\Theta_t}(s_{P_i}=k\mid\tilde{P}_i)(\boldsymbol{\mu}_k-\tilde{P}_i)}{\sum_{i=1}^{N}\mathbb{P}_{\Theta_t}(s_{P_i}=k\mid\tilde{P}_i)}.$$

By assuming that $\boldsymbol{\sigma}$ is strictly positive, we have

$$\boldsymbol{\mu}_{k,t+1}=\frac{\sum_{i=1}^{N}\tilde{P}_i\mathbb{P}_{\Theta_t}(s_{P_i}=k\mid\tilde{P}_i)}{\sum_{i=1}^{N}\mathbb{P}_{\Theta_t}(s_{P_i}=k\mid\tilde{P}_i)},\quad 0\le k\le K-1. \tag{C.5}$$

Substitute (C.5) into $\tilde{\boldsymbol{\Sigma}}_{k,t}$ to have the new covariance $\tilde{\boldsymbol{\Sigma}}^*_{k,t}$. To update $(\boldsymbol{F}_k)_{1\le k\le K}$ and $\boldsymbol{\sigma}^2$, we keep the updates $\boldsymbol{\mu}_{k,t+1}$ and introduce some latent random coefficients $(c_i)_{1\le i\le N}$ in order to perform one cycle of EM, now with respect to $(\boldsymbol{F}_k)_{0\le k\le K-1}$ and $\boldsymbol{\sigma}^2$:

$$\begin{aligned}
&\sum_{i=1}^{N}\sum_{k=0}^{K-1}\mathbb{E}_{\Theta_t}\big[\log p_\Theta(\tilde{P}_i,c_i\mid s_{P_i}=k)|\tilde{P}_i,s_{P_i}=k\big]\mathbb{P}_{\Theta_t}(s_{P_i}=k\mid\tilde{P}_i)\\
=&\sum_{i=1}^{N}\sum_{k=0}^{K-1}\mathbb{E}_{\Theta_t}\big[\log\frac{e^{-\frac{\|\tilde{P}_i-\boldsymbol{F}_kc_i-\boldsymbol{\mu}_{k,t+1}\|^2}{2\boldsymbol{\sigma}^2}-\frac{\|c_i\|^2}{2}}}{\sqrt{(2\pi)^l(2\pi\boldsymbol{\sigma}^2)^{\kappa^2}}}|\tilde{P}_i,s_{P_i}=k\big]\mathbb{P}_{\Theta_t}(s_{P_i}=k\mid\tilde{P}_i)\\
=&-\frac{1}{2}\sum_{i=1}^{N}\sum_{k=0}^{K-1}\mathbb{P}_{\Theta_t}(s_{P_i}=k\mid\tilde{P}_i)\Big(\mathbb{E}_{\Theta_t}\big[\frac{\|\tilde{P}_i-\boldsymbol{F}_kc_i-\boldsymbol{\mu}_{k,t+1}\|^2}{\boldsymbol{\sigma}^2}+\|c_i\|^2|\tilde{P}_i,s_{P_i}=k\big]\\
&\qquad\qquad+\kappa^2\log\boldsymbol{\sigma}^2+(\kappa^2+l)\log 2\pi\Big).
\end{aligned}$$

The conditional distribution of the random coefficient c_i given $\tilde{P}_i$ and $s_{P_i} = k$ is needed to compute the expectation. One way to approach the problem is to recall the best linear estimator for predicting c_i from $\tilde{P}_i$ under the k-th model:

$$c_i = L_{k,t}(\tilde{P}_i - \boldsymbol{\mu}_{k,t+1}) + m_{k,t}$$

with

1. $L_{k,t} = \boldsymbol{F}_{k,t}^T(\boldsymbol{F}_{k,t}\boldsymbol{F}_{k,t}^T + \boldsymbol{\sigma}_t^2 I_{\kappa^2})^{-1} = (\boldsymbol{F}_{k,t}^T\boldsymbol{F}_{k,t} + \boldsymbol{\sigma}_t^2 I_l)^{-1}\boldsymbol{F}_{k,t}^T := M_{k,t}^{-1}\boldsymbol{F}_{k,t}^T$;
2. $m_{k,t}$ a Gaussian vector distributed as $\mathcal{N}(0, \boldsymbol{\sigma}_t^2 M_{k,t}^{-1})$ and independent of $\tilde{P}$.

It follows

$$\begin{aligned}
&\mathbb{E}_{\Theta_t}[\boldsymbol{\sigma}^{-2}\|\tilde{P}_i - \boldsymbol{F}_k c_i - \boldsymbol{\mu}_{k,t+1}\|^2 + \|c_i\|^2|\tilde{P}_i] \\
=&\frac{\|(I_{\kappa^2} - \boldsymbol{F}_k L_{k,t})(\tilde{P}_i - \boldsymbol{\mu}_{k,t+1})\|^2 + \boldsymbol{\sigma}_t^2 tr(\boldsymbol{F}_k M_{k,t}^{-1}\boldsymbol{F}_k^T)}{\boldsymbol{\sigma}^2} + \|L_{k,t}(\tilde{P}_i - \boldsymbol{\mu}_{k,t+1})\|^2 + \boldsymbol{\sigma}_t^2 tr(M_{k,t}^{-1}) \\
=&\frac{\|\tilde{P}_i - \boldsymbol{\mu}_{k,t+1}\|^2 - 2(\tilde{P}_i - \boldsymbol{\mu}_{k,t+1})^T\boldsymbol{F}_k L_{k,t}(\tilde{P}_i - \boldsymbol{\mu}_{k,t+1}) + tr(\boldsymbol{F}_k^T\boldsymbol{F}_k\Omega_{i,k,t})}{\boldsymbol{\sigma}^2} + tr(\Omega_{i,k,t})
\end{aligned}$$

where we denote

$$\Omega_{i,k,t} = L_{k,t}(\tilde{P}_i - \boldsymbol{\mu}_{k,t+1})(\tilde{P}_i - \boldsymbol{\mu}_{k,t+1})^T L_{k,t}^T + \boldsymbol{\sigma}_t^2 M_{k,t}^{-1}$$

to further simplify the notation.

Now differentiate with respect to $\boldsymbol{F}_k$

$$\begin{aligned}
&\frac{\partial}{\partial \boldsymbol{F}_k}\Big(- 2tr(\boldsymbol{F}_k L_{k,t}(\tilde{P}_i - \boldsymbol{\mu}_{k,t+1})(\tilde{P}_i - \boldsymbol{\mu}_{k,t+1})^T) + tr(\boldsymbol{F}_k^T\boldsymbol{F}_k\Omega_{i,k,t})\Big) \\
=& - 2(\tilde{P}_i - \boldsymbol{\mu}_{k,t+1})(\tilde{P}_i - \boldsymbol{\mu}_{k,t+1})^T L_{k,t}^T + 2\boldsymbol{F}_k\Omega_{i,k,t}
\end{aligned}$$

and set the whole derivative to zero:

$$\sum_{i=1}^N \mathbb{P}_{\Theta_t}(s_{P_i} = k \mid \tilde{P}_i)\Big((\tilde{P}_i - \boldsymbol{\mu}_{k,t+1})(\tilde{P}_i - \boldsymbol{\mu}_{k,t+1})^T L_{k,t}^T - \boldsymbol{F}_k\Omega_{i,k,t}\Big) = 0.$$

It follows

$$\tilde{\boldsymbol{\Sigma}}_{k,t}^* L_{k,t}^T - \boldsymbol{F}_k(L_{k,t}\tilde{\boldsymbol{\Sigma}}_{k,t}^* L_{k,t}^T + \boldsymbol{\sigma}_t^2 M_{k,t}^{-1}) = 0$$

and

$$\boldsymbol{F}_{k,t+1} = \tilde{\boldsymbol{\Sigma}}_{k,t}^*\boldsymbol{F}_{k,t}(M_{k,t}^{-1}\boldsymbol{F}_{k,t}^T\tilde{\boldsymbol{\Sigma}}_{k,t}^*\boldsymbol{F}_{k,t} + \boldsymbol{\sigma}_t^2 I_l)^{-1}.$$

Apply the same procedure to derive an estimate for $\boldsymbol{\sigma}^2$ if it is to be updated:

$$\begin{aligned}
\boldsymbol{\sigma}_{t+1}^2 &= \frac{\sum_{i=1}^N\sum_{k=0}^{K-1}\mathbb{P}_{\Theta_t}(s_{P_i} = k \mid \tilde{P}_i)\Big(\|(I_{\kappa^2} - \boldsymbol{F}_{k,t+1}L_{k,t})(\tilde{P}_i - \boldsymbol{\mu}_{k,t+1})\|^2 + \boldsymbol{\sigma}_t^2 tr\big(\boldsymbol{F}_{k,t+1}M_{k,t}^{-1}\boldsymbol{F}_{k,t+1}^T\big)\Big)}{\kappa^2\cdot\sum_{i=1}^N\sum_{k=0}^{K-1}\mathbb{P}_{\Theta_t}(s_{P_i} = k \mid \tilde{P}_i)} \\
&= \frac{\sum_{k=0}^{K-1} tr\Big(\tilde{\boldsymbol{\Sigma}}_{k,t}^* - \tilde{\boldsymbol{\Sigma}}_{k,t}^*\boldsymbol{F}_{k,t}M_{k,t}^{-1}\boldsymbol{F}_{k,t+1}^T\Big)\sum_{i=1}^N\mathbb{P}_{\Theta_t}(s_{P_i} = k \mid \tilde{P}_i)}{\kappa^2 N}
\end{aligned}$$

because

$$\sum_{i=1}^{N} \mathbb{P}_{\Theta_t}(s_{P_i} = k \mid \tilde{P}_i) \|(I_{\kappa^2} - \boldsymbol{F}_{k,t+1} L_{k,t})(\tilde{P}_i - \boldsymbol{\mu}_{k,t+1})\|^2$$

$$= tr\big(\tilde{\boldsymbol{\Sigma}}^*_{k,t}(I_{\kappa^2} - \boldsymbol{F}_{k,t+1}L_{k,t} - L^T_{k,t}\boldsymbol{F}^T_{k,t+1} + L^T_{k,t}\boldsymbol{F}^T_{k,t+1}\boldsymbol{F}_{k,t+1}L_{k,t})\big) \sum_{i=1}^{N} \mathbb{P}_{\Theta_t}(s_{P_i} = k \mid \tilde{P}_i)$$

and

$$tr\big(\boldsymbol{F}^T_{k,t+1}\boldsymbol{F}_{k,t+1}(L_{k,t}\tilde{\boldsymbol{\Sigma}}^*_{k,t}L^T_{k,t} + \boldsymbol{\sigma}^2_t M^{-1}_{k,t})\big) = tr\big(\boldsymbol{F}^T_{k,t+1}\tilde{\boldsymbol{\Sigma}}^*_{k,t}L^T_{k,t}\big) = tr\big(\tilde{\boldsymbol{\Sigma}}^*_{k,t}\boldsymbol{F}_{k,t+1}L_{k,t}\big).$$

Appendix D. Information Theoretic Interpretation of Probabilistic PCA.

Let X be a n-dimensional Gaussian vector distributed as $\mathcal{N}(\mu, \Sigma)$. Given a set of covariance matrices $\mathfrak{C} = \{FF^T + \sigma^2 I,\ F \in \mathbb{R}^{n\times l}, \sigma \in \mathbb{R}_+\}$, one seeks the element in $\mathfrak{C}$ that minimizes the Kullback-Leibler divergence between the two probabilities

$$(F_*, \sigma_*) = \operatorname*{argmin}_{F\in\mathbb{R}^{n\times l},\sigma\in\mathbb{R}_+} \mathbb{E}\Big[\log \frac{p_{\mu,\Sigma}(X)}{p_{\mu,FF^T+\sigma^2 I}(X)}\Big]. \tag{D.1}$$

Note that the expectation in (D.1) is defined under the probability induced by X, a choice intended to simplify the calculation:

$$\begin{aligned}
&\operatorname*{argmin}_{F\in\mathbb{R}^{n\times l},\sigma\in\mathbb{R}_+} \mathbb{E}\Big[\log \frac{p_{\mu,\Sigma}(X)}{p_{\mu,FF^T+\sigma^2 I}(X)}\Big] \\
=& \operatorname*{argmax}_{F\in\mathbb{R}^{n\times l},\sigma\in\mathbb{R}_+} \mathbb{E}\Big[\log p_{\mu,FF^T+\sigma^2 I}(X)\Big] \\
=& \operatorname*{argmax}_{F\in\mathbb{R}^{n\times l},\sigma\in\mathbb{R}_+} \mathbb{E}\Big[-\frac{1}{2}\log\det(FF^T+\sigma^2 I) - \frac{1}{2}(X-\mu)^T(FF^T+\sigma^2 I)^{-1}(X-\mu)\Big] \\
=& \operatorname*{argmin}_{F\in\mathbb{R}^{n\times l},\sigma\in\mathbb{R}_+} \log\det(FF^T+\sigma^2 I) + tr\Big[(FF^T+\sigma^2)^{-1}\mathbb{E}[(X-\mu)(X-\mu)^T]\Big] \\
=& \operatorname*{argmin}_{F\in\mathbb{R}^{n\times l},\sigma\in\mathbb{R}_+} \log\det(FF^T+\sigma^2 I) + tr\Big[(FF^T+\sigma^2)^{-1}\Sigma\Big].
\end{aligned}$$

Appendix E. Stationarity and Convergence.

Let us denote $\mathfrak{N}_N := \mathbb{Z} \cap [1, N]$ in what follows.

Theorem E.1. *Let $(X_{t,s})_{(t,s)\in\mathbb{Z}^2}$ be a real-valued process with its mean and covariance defined by*

$$\forall (t,s), (t',s') \in \mathbb{Z}^2,\ \ \mathbb{E}[X_{t,s}] = \mu,\ \ cov[X_{t,s}, X_{t',s'}] = R_X(t-t', s-s')$$

where μ is a constant and $R_X(\cdot,\cdot)$ satisfies

$$\sum_{(t,s)\in\mathbb{Z}^2} |R_X(t,s)| < +\infty.$$

Then the average $N^{-2}\sum_{(t,s)\in\mathfrak{N}_N^2} X_{t,s}$ *converges both almost surely and in* $\mathbb{L}^2$ *to* μ *as* N *goes to infinity.*

Proof. With the random variable $N^{-2}\sum_{(t,s)\in\mathfrak{N}_N^2} X_{t,s}$ denoted by A_N, we have

$$\mathbb{E}[\|A_N-\mu\|^2] = N^{-4}\mathbb{E}[\|\sum_{(t,s)\in\mathfrak{N}_N^2}(X_{t,s}-\mu)\|^2] = N^{-4}\sum_{\substack{(t,s)\in\mathfrak{N}_N^2\\(t',s')\in\mathfrak{N}_N^2}} R_X(t-t',s-s').$$

Lesbesgue's dominated convergence theorem then implies

$$\lim_{N\to\infty} N^{-2}\sum_{\substack{(t,s)\in\mathfrak{N}_N^2\\(t',s')\in\mathfrak{N}_N^2}} R_X(t-t',s-s') = \sum_{(t,s)\in\mathbb{Z}^2} R_X(t,s)$$

and the convergence in $\mathbb{L}^2$

$$\lim_{N\to\infty}\mathbb{E}[\|A_N-\mu\|^2]=0$$

follows from

$$\mathbb{E}[\|A_N-\mu\|^2] \le N^{-2}\sum_{(t,s)\in\mathbb{Z}^2}|R_X(t,s)|. \tag{E.1}$$

In addition, Markov's inequality implies

$$\forall c>0,\ \ \mathbb{E}[\|A_N-\mu\|^2] \ge c^2\mathbb{P}(|A_N-\mu|\ge c)$$

which, combined with (E.1), leads to

$$\forall c>0,\ \ N^{-2}\sum_{(t,s)\in\mathbb{Z}^2}|R_X(t,s)| \ge c^2\mathbb{P}(|A_N-\mu|\ge c)$$

and thus

$$\forall c>0,\ \ \sum_{N=1}^{+\infty}\mathbb{P}(|A_N-\mu|\ge c) < +\infty.$$

The almost sure convergence follows from the Borel-Cantelli lemma. ∎

The condition

$$\sum_{(t,s)\in\mathbb{Z}^2}|R_X(t,s)|<+\infty$$

requires $R_X(\cdot,\cdot)$ to decay at a certain rate so that instances of $X_{t,s}$ from two sites far from each other do not behave in sync. In view of the proof, this condition can be weakened to

$$\exists\alpha\in[0,1),\ \limsup_{N\to\infty} N^{-\alpha}\sum_{(t,s)\in\mathfrak{N}_N^2}|R_X(t,s)|<+\infty.$$

As an application, assume for convenience an infinitely large image in which the overlapping patches $(P^{(p,q)})_{(p,q)\in\mathbb{Z}^2}$ form a stationary process valued in $\mathbb{R}^{\kappa^2}$. Thus the noisy patches $(\tilde{P}^{(p,q)})_{(p,q)\in\mathbb{Z}^2}$, seen as a process, are stationary too because the set of stationary processes is closed under addition. The same property can then be extended to $g(P^{(p,q)}, \tilde{P}^{(p,q)})$ where $g : \mathbb{R}^{\kappa^2} \times \mathbb{R}^{\kappa^2} \mapsto \mathbb{R}$ is square integrable under the probability induced by the random couple $(P^{(p,q)}, \tilde{P}^{(p,q)})$. For example, $\|P^{(p,q)} - f(\tilde{P}^{(p,q)})\|^2$ if we take f to mean any of the filters mentioned in theorem 3.2. Its integrability can be easily checked

$$\mathbb{E}[\|P^{(p,q)} - f(\tilde{P}^{(p,q)})\|^4] \leq \mathbb{E}[(\|P^{(p,q)}\| + \|f\|\|\tilde{P}^{(p,q)}\|)^4] \leq 8(\mathbb{E}[\|P^{(p,q)}\|^4] + \|f\|^4\mathbb{E}[\|\tilde{P}^{(p,q)}\|^4])$$

where the first inequality follows from the triangular inequality and the second results from Jensen's inequality ($\|f\|$ denotes the bounded operator's norm). Now we can define

$$\forall (p,q),(p',q') \in \mathbb{Z}^2, \quad R_f(p-p', q-q') := cov[\|P^{(p,q)} - f(\tilde{P}^{(p,q)})\|^2, \|P^{(p',q')} - f(\tilde{P}^{(p',q')})\|^2].$$

Assume further that $R_f(\cdot,\cdot)$ belongs to $\mathbb{L}^1(\mathbb{Z}^2)$ under the counting measure, which is not unreasonable given the restricted support of image patches. Theorem E.1 then implies that the empirical average of $\|P^{(p,q)} - f(\tilde{P}^{(p,q)})\|^2$ converges in two senses simultaneously to its mean as sample size grows to infinity.

REFERENCES

[1] M. Aharon, M. Elad, and A. Bruckstein, *K-svd: Design of dictionaries for sparse representation*, Proceedings of SPARS, 5 (2005), pp. 9–12.

[2] PJ Brockwell and RA Davis, *Time series: Theory and models*, 1991.

[3] A. Buades, B. Coll, and J.M. Morel, *A review of image denoising algorithms, with a new one*, Multiscale Modeling & Simulation, 4 (2005), pp. 490–530.

[4] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel, *Non-Local Means Denoising*, Image Processing On Line, 2011 (2011). http://dx.doi.org/10.5201/ipol.2011.bcm_nlm.

[5] P. Chatterjee and P. Milanfar, *Clustering-based denoising with locally learned dictionaries*, Image Processing, IEEE Transactions on, 18 (2009), pp. 1438–1451.

[6] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, *Image restoration by sparse 3d transform-domain collaborative filtering*, in SPIE Electronic Imaging, vol. 6812, Citeseer, 2008.

[7] A.P. Dempster, N.M. Laird, and D.B. Rubin, *Maximum likelihood from incomplete data via the em algorithm*, Journal of the Royal Statistical Society. Series B (Methodological), (1977), pp. 1–38.

[8] D.L. Donoho and I.M. Johnstone, *Adapting to unknown smoothness via wavelet shrinkage*, Journal of the american statistical association, 90 (1995), pp. 1200–1224.

[9] D.L. Donoho and J.M. Johnstone, *Ideal spatial adaptation by wavelet shrinkage*, Biometrika, 81 (1994), pp. 425–455.

[10] A.A. Efros and T.K. Leung, *Texture synthesis by non-parametric sampling*, in Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on, vol. 2, IEEE, 1999, pp. 1033–1038.

[11] M. Elad and M. Aharon, *Image denoising via sparse and redundant representations over learned dictionaries*, Image Processing, IEEE Transactions on, 15 (2006), pp. 3736–3745.

[12] C. Harris and M. Stephens, *A combined corner and edge detector*, in Alvey vision conference, vol. 15, Manchester, UK, 1988, p. 50.

[13] A.O. Hero and J.A. Fessler, *Convergence in norm for alternating expectation-maximization (em) type algorithms*, Statistica Sinica, 5 (1995), pp. 41–54.

[14] C. Kervrann and J. Boulanger, *Optimal spatial adaptation for patch-based image denoising*, Image Processing, IEEE Transactions on, 15 (2006), pp. 2866–2878.

[15] ———, *Local adaptivity to variable smoothness for exemplar-based image regularization and representation*, International Journal of Computer Vision, 79 (2008), pp. 45–69.

[16] MARC LEBRUN, *An Analysis and Implementation of the BM3D Image Denoising Method*, Image Processing On Line, 2012 (2012). http://dx.doi.org/10.5201/ipol.2012.l-bm3d.

[17] MARC LEBRUN, ANTONI BUADES, AND JEAN-MICHEL MOREL, *Implementation of the "Non-Local Bayes" Image Denoising Algorithm (preprint)*, Image Processing On Line, (2012).

[18] MARC LEBRUN AND ARTHUR LECLAIRE, *An Implementation and Detailed Analysis of the K-SVD Image Denoising Algorithm*, Image Processing On Line, 2012 (2012). http://dx.doi.org/10.5201/ipol.2012.llm-ksvd.

[19] A. LEVIN AND B. NADLER, *Natural image denoising: Optimality and inherent bounds*, in Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE, 2011, pp. 2833–2840.

[20] P. MILANFAR, *A tour of modern image processing*, IEEE Signal Process Mag, (2011).

[21] J. PORTILLA, V. STRELA, M.J. WAINWRIGHT, AND E.P. SIMONCELLI, *Image denoising using scale mixtures of gaussians in the wavelet domain*, Image Processing, IEEE Transactions on, 12 (2003), pp. 1338–1351.

[22] BOSHRA RAJAEI, *Study and analysis of GSM (preprint)*, Image Processing On Line, (2012).

[23] S. ROWEIS, *Em algorithms for pca and spca*, Advances in neural information processing systems, (1998), pp. 626–632.

[24] L.I. RUDIN, S. OSHER, AND E. FATEMI, *Nonlinear total variation based noise removal algorithms*, Physica D: Nonlinear Phenomena, 60 (1992), pp. 259–268.

[25] L. SHAPIRO AND G.C. STOCKMAN, *Computer vision. 2001*, 2001.

[26] C.M. STEIN, *Estimation of the mean of a multivariate normal distribution*, The annals of Statistics, (1981), pp. 1135–1151.

[27] H. TAKEDA, S. FARSIU, AND P. MILANFAR, *Kernel regression for image processing and reconstruction*, Image Processing, IEEE Transactions on, 16 (2007), pp. 349–366.

[28] M.E. TIPPING AND C.M. BISHOP, *Mixtures of probabilistic principal component analyzers*, Neural computation, 11 (1999), pp. 443–482.

[29] ———, *Probabilistic principal component analysis*, Journal of the Royal Statistical Society: Series B (Statistical Methodology), 61 (1999), pp. 611–622.

[30] YI-QING WANG, *The Implementation of SURE Guided Piecewise Linear Denoising (preprint)*, Image Processing On Line, (2012).

[31] GUOSHEN YU AND GUILLERMO SAPIRO, *DCT image denoising: a simple and effective image denoising algorithm*, Image Processing On Line, 2011 (2011). http://dx.doi.org/10.5201/ipol.2011.ys-dct.

[32] G. YU, G. SAPIRO, AND S. MALLAT, *Solving inverse problems with piecewise linear estimators: from gaussian mixture models to structured sparsity*, Image Processing, IEEE Transactions on, 21 (2012), pp. 2481–2499.

[33] D. ZORAN AND Y. WEISS, *From learning models of natural image patches to whole image restoration*, in Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE, 2011, pp. 479–486.