IPOL IMAGE PROCESSING ON LINE

Published in Image Processing On Line on 2013–07–01.
Submitted on 2012–11–25, accepted on 2013–02–13.
ISSN 2105–1232 © 2013 IPOL & the authors CC–BY–NC–SA
This article is available online with supplementary materials,
software, datasets and online demo at
https://doi.org/10.5201/ipol.2013.52

The Implementation of SURE Guided Piecewise Linear Image Denoising

Yi-Qing Wang[1]

[1] CMLA, ENS Cachan, France (yiqing.wang@cmla.ens-cachan.fr)

Communicated by Mauricio Delbracio *Demo edited by* Yi-Qing Wang

This IPOL article is related to a companion publication in the SIAM Journal on Imaging Sciences:
Y.Q. Wang and J.M. Morel. "SURE Guided Gaussian Mixture Image Denoising". SIAM Journal on Imaging Sciences, 2013.
http://dx.doi.org/10.1137/120901131

Abstract

SURE (Stein's Unbiased Risk Estimator) guided Piecewise Linear Estimation (S-PLE) is a recently introduced patch-based state-of-the-art denoising algorithm. In this article, we focus on its implementation and show its performance by comparing it with several other acclaimed algorithms.

Source Code

ANSI C source code for both S-PLE and PLE is accessible on the article web page. A live demo for S-PLE can be found at the IPOL web page of this article[1].

Keywords: denoising, expectation-maximization, Stein's unbiased risk estimator

1 Introduction

A novel patch-based image denoising algorithm, SURE guided Piecewise Linear Estimation (S-PLE), is presented by Wang et al. [25]. It assumes that 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. Before delving

[1] https://doi.org/10.5201/ipol.2013.52

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 $(\boldsymbol{w}_k)_{0\le k\le K-1}$ with $\sum_{k=0}^{K-1} \boldsymbol{w}_k = 1$, vectors $(\boldsymbol{\mu}_k)_{0\le k\le K-1}$, and positive semidefinite matrices $(\boldsymbol{\Sigma}_k)_{0\le k\le K-1}$ 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.

The observation model for image denoising under GMM is

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

with s_P, the patch model selector, a discrete random variable distributed according to $(\boldsymbol{w}_k)_{0\le k\le K-1}$ and independent of the noise term N. And the conditional expectation of P given $\tilde{P}$, which constitutes the optimal filter in the $\mathbb{L}^2$ sense, has a closed form

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

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

The field has known significant progress in the last few decades. DCT [26] showcases the versatility of the shrinkage [9] when combined with a good basis. BLS-GSM [17] illustrates the power of natural image statistics modeling in the wavelet domain. Non-Local Means (NLM) [3, 4], inspired in part by the pioneering work by Efros et al. [10] in texture synthesis, effectively exploits information redundancy in natural images. Through similar patch grouping and collaborative filtering, BM3D [6, 14] further enhanced NLM and catapulted it to one of the best performing denoising methods that define the current state-of-the-art. Non-Local Bayes (NLBayes) [16] in turn improves BM3D by aggressively going after flat areas in an image and largely addresses its tendency to create artifacts in strong noise.

Another promising direction of research initiated by Aharon et al. [1, 11] proposed a greedy orthogonal matching pursuit algorithm based on the notion that image patches can be sparsely represented with an over-complete dictionary. Then a patch orientation based dictionary learning algorithm [5] gave rise to the K-LLD denoising algorithm. Later an algorithm called PLE [27] was designed along a similar vein, but intended as a generic image recovery related inverse problem solver. In a recent development Zoran et al. [28] introduced a new optimization scheme which continued the effort started as early as in 1992 [20] of seeking an adequate description of image priors. Instead of constructing priors for images as a whole, a prior for image patches in the form of a Gaussian mixture was constructed and produced impressive results.

S-PLE, unlike those methods spawned by the NLM paradigm, groups image patches by assessing patch-to-model rather than patch-to-patch distance. As discussed 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 [22] 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.

2 PLE

In this section, PLE [27] is described to highlight its difference with S-PLE. PLE starts with building a number of directional models using synthetic samples and it retains all the eigenvectors from the estimated covariance matrices. Then one additional model is constructed using DCT as its basis to account for textural patches. Contrary to S-PLE, the model means and their covariance eigenvalues are arbitrarily fixed (see algorithm 1).

Algorithm 1 PLE initialization

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 $\{(r, u) : B(r, u) = 0\}$ and $\{(r, u) : B(r, u) = 255\}$ separated by a straight line inclined at $\frac{k}{K-1}\pi$ passing through the center of the image.
2. Blur B with Gaussian kernels of different standard deviations $(\sigma_b)_{1 \le b \le 4}$: $\sigma_b = 2b$ for all b.
3. Draw $100\kappa^2$ $\kappa \times \kappa$ 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. Define 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.[2]

end for
To this setup add a textural model whose basis is formed by DCT (with ascending component frequencies). Set its 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 just built.

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

Assume that there are K models in all. For each patch to restore, PLE produces K estimates under individual model assumption and keeps the one with the highest conditional probability to have both the observation and its estimate. This patch is assigned in the meantime to the same model.

Finally, all the models are updated with their assigned estimates. The last two steps, called estimation and maximization by the paper, are then repeated several times before the algorithm terminates (see algorithm 2).

Algorithm 2 PLE

Input: A noisy gray image $\tilde{U}$, its noise standard deviation σ.
Parameter: Number of PLE iterations S.
Output: Denoised image.
Run algorithm 1. Extract all $\kappa \times \kappa$ patches from $\tilde{U}$ to have $(\tilde{P}_i)_{1 \leq i \leq N}$.
for $t = 1$ to S **do**
Estimation:

1. Maximize the conditional density given the observation and the model with (2):

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

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

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

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 $\epsilon = 10^{-3}$ to ensure the definiteness of $\boldsymbol{\Sigma}_{k,t}$.
end for
end for
Assign equal weights to all restored patches and recover the image. A typical pixel inside the image will hence be the arithmetic average of all its κ^2 estimates.

The quadratic minimization problem

$$\widehat{P} = \underset{P}{\operatorname{argmin}} \left(\frac{\|P - \tilde{P}\|^2}{\sigma^2} + (P - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (P - \boldsymbol{\mu}_k) \right)$$

in the algorithm's estimation step is solved by

$$\begin{aligned} \widehat{P} &= \sigma^{-2}(\sigma^{-2} I + \boldsymbol{\Sigma}_k^{-1})^{-1}(\tilde{P} - \boldsymbol{\mu}_k) + \boldsymbol{\mu}_k \\ &= \boldsymbol{\Sigma}_k(\sigma^2 I + \boldsymbol{\Sigma}_k)^{-1}(\tilde{P} - \boldsymbol{\mu}_k) + \boldsymbol{\mu}_k \end{aligned} \tag{2}$$

with I representing the identity of the same dimension as $\boldsymbol{\Sigma}_k$ and σ the noise level. Hence we find the Wiener filter expression as expected because if signal and noise follow a joint Gaussian distribution, the least square estimator is known to be the maximum likelihood estimator, too.

PLE is not GMM-based for it involves no concept of model priors. Moreover, no criterion in PLE is guaranteed to converge in any sense, making it difficult to decide when to stop the algorithm. And there is too much latitude in tuning parameters such as the model eigenvalues, which often undermines the algorithm's numerical stability and performance. In addition, 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.

3 SURE Guided PLE Denoising Algorithm

3.1 Gaussian Factor Mixture

There is reason to believe 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 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. In this implementation, 18 such oriented models together with two *non-oriented* components, representing textural and flat patches respectively, were set up in the mixture.

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 [23] and adopted in this implementation.

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 by Yu et al. [27], 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] (implemented in `sample_images()`): 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

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

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

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

is small, where λ_b and λ_s ($\lambda_b \geq \lambda_s \geq 0$) are the two eigenvalues of M_P. Thus a threshold $t_{\textbf{orient}} = 5$ was tuned according to our subjective view so that a patch satisfying $\lambda_s^{-1}\lambda_b \geq t_{\textbf{orient}}$ 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 $\psi(a) = a 1_{a\geq 0} + (\pi + a) 1_{a<0}$, the latter of which ensures the positivity of θ_*.

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

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

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

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

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 $\lambda_s^{-1}\lambda_b \geq t_{\textbf{orient}}$ 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

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

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

We 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 figure 1 are three resulting covariance matrices. Several observations are in order:

1. 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.

2. 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 decay 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;

3. third, to prevent over-fitting, the first leading eigenvector was made the sole factor representing the flat model. As a reflection of the richness of textural content, the number of components in the textural model was set to 63: a DCT-like isotropic basis thus breaks up into two to handle two radically different patch categories.

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:

$$\begin{aligned}
(\boldsymbol{F}_*, \boldsymbol{\sigma}_*) &= \underset{\boldsymbol{F}\in\mathbb{R}^{\kappa^2\times l}, \boldsymbol{\sigma}\in\mathbb{R}_+}{\operatorname{argmax}} \log \prod_{i=1}^{N} \frac{1}{\sqrt{(2\pi)^{\kappa^2}\det(\boldsymbol{F}\boldsymbol{F}^T + \boldsymbol{\sigma}^2 I)}} \exp\Big(-\frac{1}{2}(P_i-\mu)^T(\boldsymbol{F}\boldsymbol{F}^T + \boldsymbol{\sigma}^2 I)^{-1}(P_i-\mu)\Big) \\
&= \underset{\boldsymbol{F}\in\mathbb{R}^{\kappa^2\times l}, \boldsymbol{\sigma}\in\mathbb{R}_+}{\operatorname{argmin}} \frac{N}{2}\Big[\log\det(\boldsymbol{F}^T\boldsymbol{F} + \boldsymbol{\sigma}^2 I) + \frac{1}{N}\sum_{i=1}^{N} tr\Big((\boldsymbol{F}\boldsymbol{F}^T + \boldsymbol{\sigma}^2 I)^{-1}(P_i-\mu)(P_i-\mu)^T\Big)\Big] \\
&= \underset{\boldsymbol{F}\in\mathbb{R}^{\kappa^2\times l}, \boldsymbol{\sigma}\in\mathbb{R}_+}{\operatorname{argmin}} \log\det(\boldsymbol{F}\boldsymbol{F}^T + \boldsymbol{\sigma}^2 I) + tr\Big((\boldsymbol{F}\boldsymbol{F}^T + \boldsymbol{\sigma}^2 I)^{-1}\Sigma\Big). \qquad (3)
\end{aligned}$$

The problem has been dealt with and lead to probabilistic PCA [23, 24, 19].

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

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

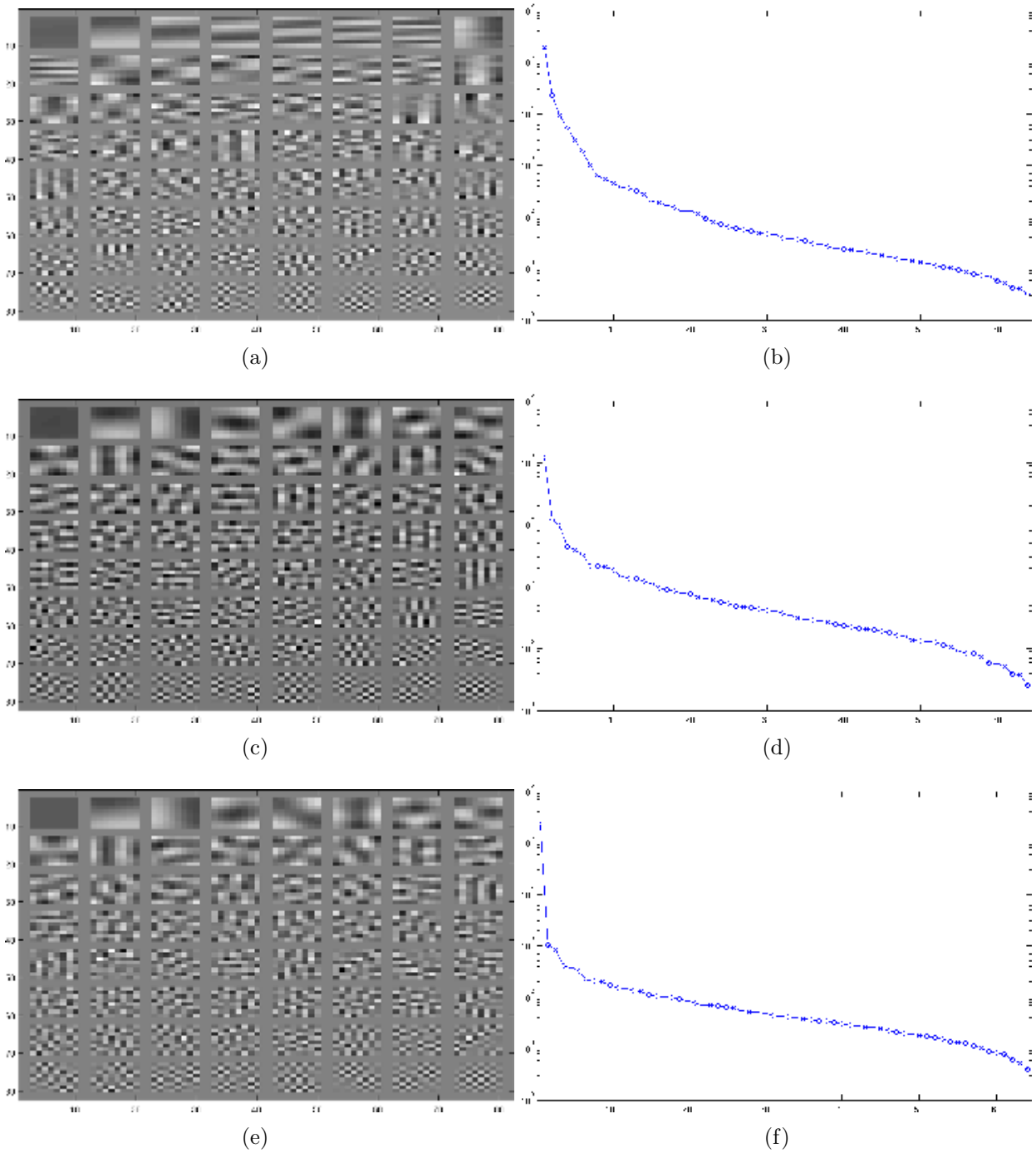


Figure 1: Examples of the eigenvectors and eigenvalues obtained by sampling 493 gray images from the *Berkeley Segmentation Dataset* with $K = 20$: a patch view of the eigenvectors of the (a) 0-th (mono-oriented), (c) 18-th (multi-oriented), (e) 19-th (flat) cluster and their eigenvalues displayed in the logarithmic scale: the (b) 0-th, (d) 18-th and (f) 19-th cluster.

with N_k the total number of patches collected for the k-th model satisfying $\min_{0\leq k\leq K-1} N_k = 5000$. As shown in figure 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 same mixture prior was used in all our experiments.

The complete algorithm for GFM initialization of S-PLE is described in algorithm 3.

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 the color image *dice* from the IPOL website and added to its color channels some simulated i.i.d. zero-mean Gaussian noise with standard deviation equal to 10.

To the 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:

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

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.

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 $\{\mathbb{P}(s_P = k \mid \tilde{P}),\ 0 \leq k \leq 19\}$, which allowed us to determine the most suitable model for $\tilde{P}$ simply by comparing its likelihoods under different model assumptions:

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

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 (figure 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.

Algorithm 3 GMM initialization of S-PLE

Input: Z noiseless natural gray images.
Parameter: Number of mixture components K, patch dimension $\kappa \times \kappa$.
Output: K Gaussian mixture components and their mixing weights.
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

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

which is the solution to the optimization problem (3).
end for

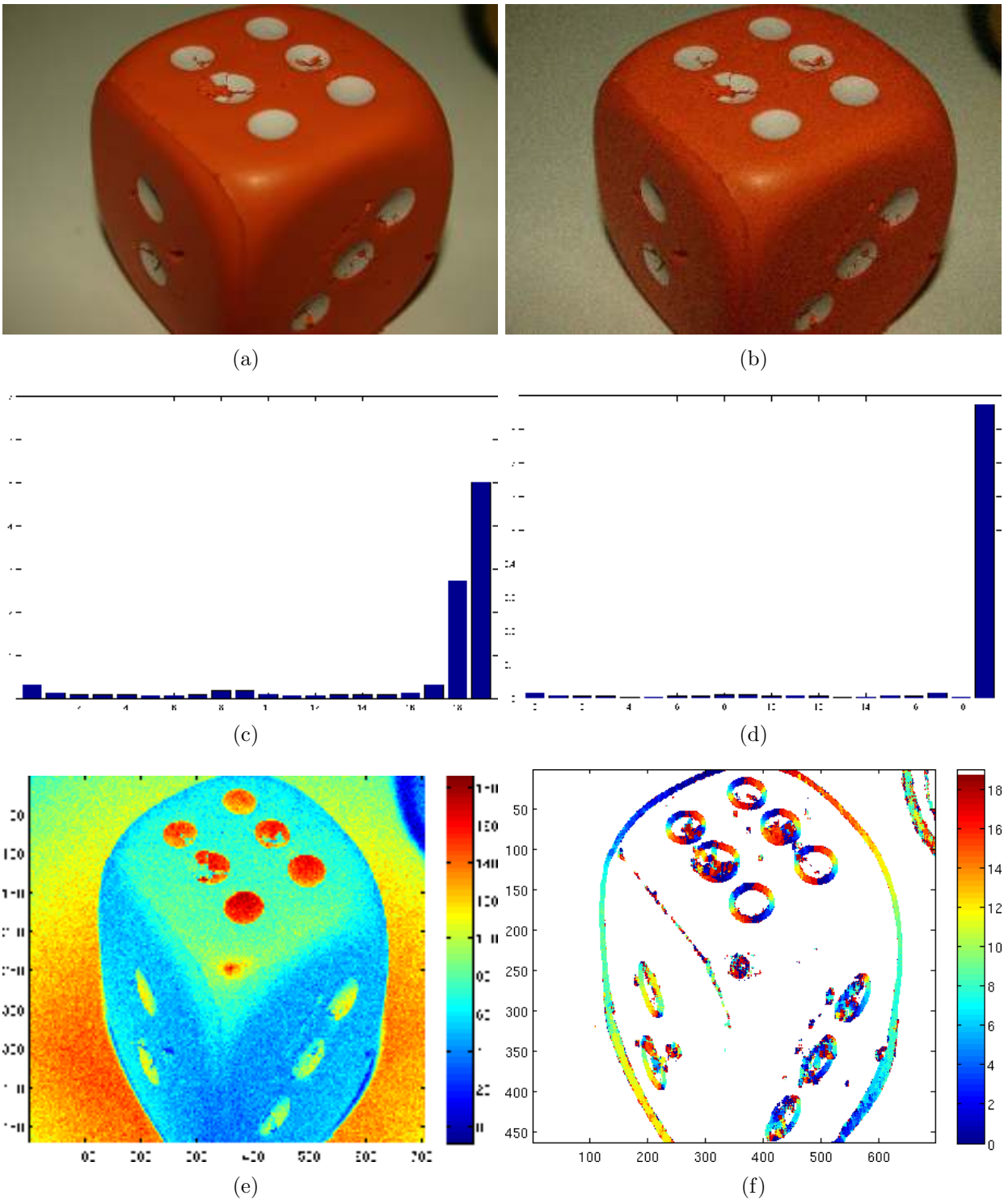


Figure 2: (a) original image (b) noisy image ($\sigma = 10$) (c) initial model priors (d) updated model priors after the first EM iteration on the first transformed channel (e) pixel-wise arithmetic mean of the noisy image's three color channels (f) patch map formed after the first EM iteration. White color highlights the patches classified as essentially flat.

3.4 SURE-Aided Adaptive Filtering

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 K-LLD [5]. On the other hand, although the constantly increasing overall likelihood is an 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, despite 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 [28] attempted to reconcile these two concerns by tying them together to form a single cost function. We address the problem with the help of SURE [22] 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 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:* $f(\tilde{P}) = \sum_{j=1}^{\kappa^2} c_j \langle \tilde{P} - \mu, b_j \rangle b_j + \mu$
2. *soft shrinkage:* $f(\tilde{P}) = \sum_{j=1}^{\kappa^2} c_j \gamma_t^{\boldsymbol{soft}}\big(\langle \tilde{P} - \mu, b_j \rangle\big) b_j + \mu$ *with* $\gamma_t^{\boldsymbol{soft}}(\omega) = sgn(\omega)(|\omega| - t)_+$
3. *hard shrinkage:* $f(\tilde{P}) = \sum_{j=1}^{\kappa^2} c_j \gamma_t^{\boldsymbol{hard}}\big(\langle \tilde{P} - \mu, b_j \rangle\big) b_j + \mu$ *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:* $\nabla \cdot f(\tilde{P}) = \sum_{j=1}^{\kappa^2} c_j 1_{[t,+\infty)}\big(\big|\langle \tilde{P} - \mu, b_j \rangle\big|\big)$
3. *hard shrinkage:* $\nabla \cdot f(\tilde{P}) = \sum_{j=1}^{\kappa^2} c_j \big(1_{[t,+\infty)}\big(\big|\langle \tilde{P} - \mu, b_j \rangle\big|\big) + t\mathbb{E}\big[(\delta_t - \delta_{-t})\big(\langle \tilde{P} - \mu, b_j \rangle\big) \mid P\big]\big)$

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

Theorem 1 *Under the assumptions in Definition 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

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

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 density difference at t and $-t$

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

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

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

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

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

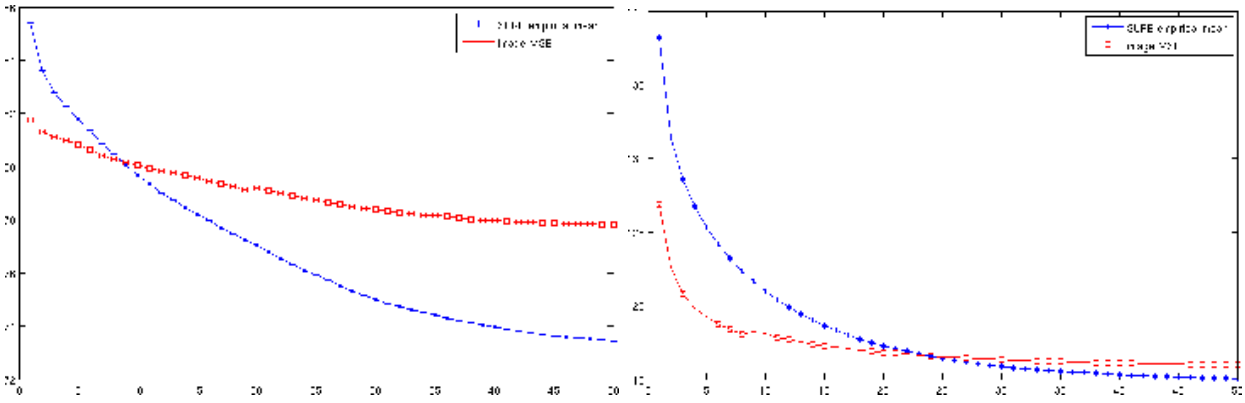
is an unbiased estimator of the expected patch MSE $\kappa^{-2}\mathbb{E}[\|P - f(\tilde{P})\|^2]$ and it converges

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

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

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 for 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) 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.

Perhaps more interestingly, this SURE empirical mean can be shown as an asymptotic upper bound on the MSE of the restored image. Hence we have one more reason to monitor it 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 (figure 3).



(a) (b)

Figure 3: (a) $\sigma = 20$: the MSE of the restored *traffic* image and their corresponding SURE empirical mean at each S-PLE iteration (b) $\sigma = 20$: the MSE of the restored *valldemossa* image 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 explained by the calculated SURE being biased because of the explained approximation used in dealing with non-linear Wiener filtering coefficients.

3.4.2 Self-adjusting Flat Patches

We settled for the 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. When denoising a flat 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 both patches are the same shall be upheld statistically;

The first condition can be easily checked thanks to the patch map and the connected component labeling algorithm [3] [21] 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 $\mathbb{P}_{test}$. To take into account this possibly overly simplified null hypothesis and ensure a high likelihood for retaining at least one additional patch to help denoise P, the chi-square test threshold $\mathfrak{t} = 65$ is thus set to verify

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

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. It ought to be mentioned that when noise is strong, EM can mistake weak borders for noise and cause some patch orientation to be not properly recognized, which usually happens in the areas of subtle and gradual color transition (figure 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.

[3] We obtained a version from http://alumni.media.mit.edu/~rahimi/connected/.

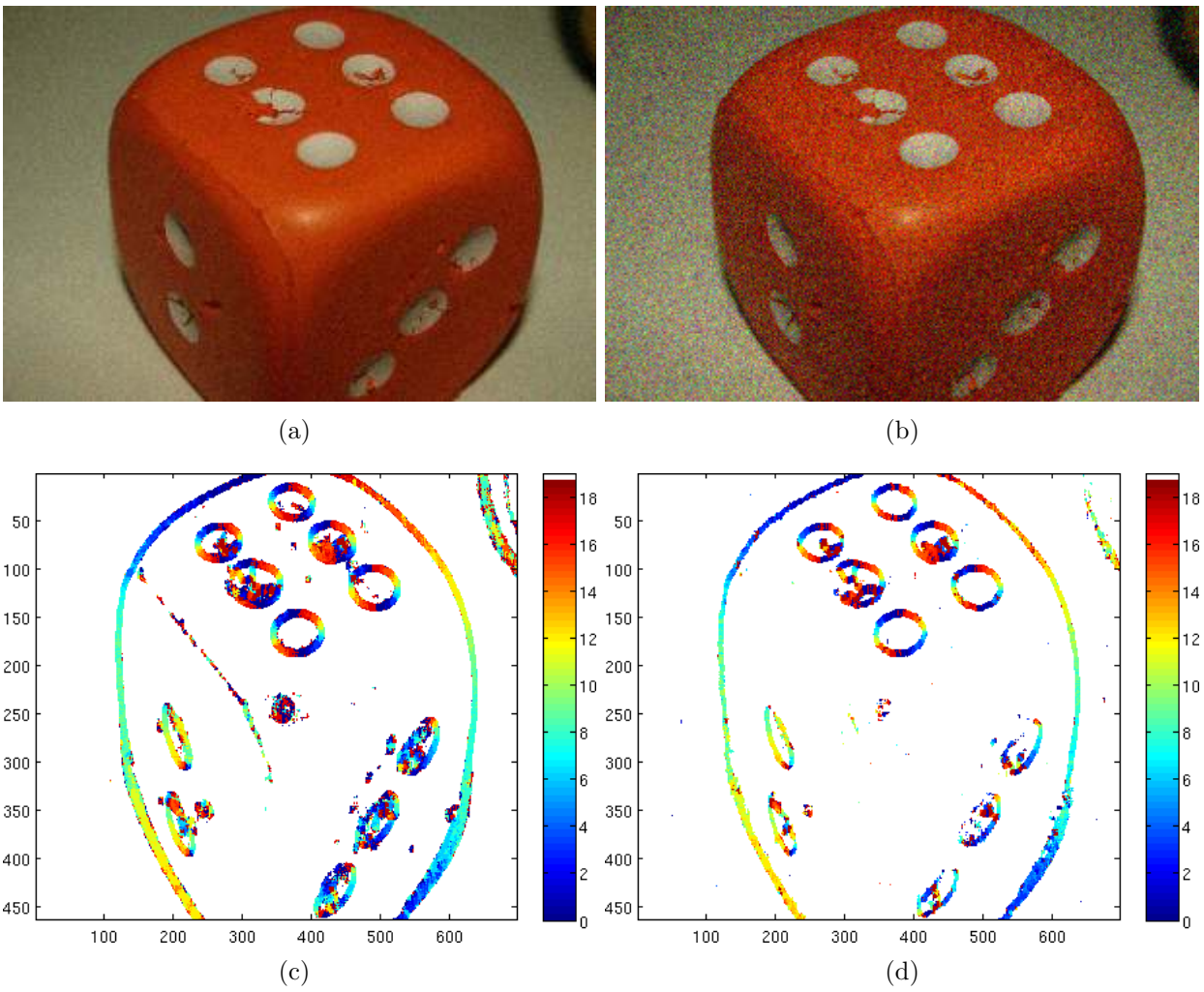


Figure 4: The two images in the first row are *dice* degraded by Gaussian noise with (a) $\sigma = 10$ and (b) $\sigma = 30$ respectively. EM iterated twice on the first transformed channel as explained in figure 2 to produce the patch map for (c) $\sigma = 10$ and (d) $\sigma = 30$. Notice that the oriented edge on the top side of the dice failed to be recognized at $\sigma = 30$.

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 (algorithm 4): 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. 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.

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 $\mathfrak{t}$.
Output: Potentially updated flat patches estimated with an expanded support.
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

4 Algorithm Outline and a Comparative Study

The complete S-PLE algorithm is described in algorithm 5.

To reduce execution time when denoising color images, instead of running the computationally intensive EM algorithm on the 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 thus 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 observation

Algorithm 5 S-PLE

Input: A noisy gray image $\tilde{U}$.
Parameter: Number of EM iteration S, noise level σ.
Output: Denoised image.
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 $\forall \tilde{P} \in \mathcal{P},\ \forall 0 \leq k \leq 19,\ \mathbb{P}_{\Theta_0}(s_P = k \mid \tilde{P})$.
for $t = 1$ to S **do**
Update model priors:

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

Update model means:

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

Update factor loadings:

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

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

$$\forall 0 \leq k \leq 19, \quad M_{k,t-1} = \boldsymbol{F}^T_{k,t-1} \boldsymbol{F}_{k,t-1} + \sigma^2 I_{l_k} \qquad \text{and}$$

$$\tilde{\boldsymbol{\Sigma}}^*_{k,t-1} = \frac{\sum_{\tilde{P} \in \mathcal{P}} (\tilde{P} - \boldsymbol{\mu}_{k,t})(\tilde{P} - \boldsymbol{\mu}_{k,t})^T \mathbb{P}_{\Theta_{t-1}}(s_P = k \mid \tilde{P})}{\sum_{\tilde{P} \in \mathcal{P}} \mathbb{P}_{\Theta_{t-1}}(s_P = k \mid \tilde{P})}.$$

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:

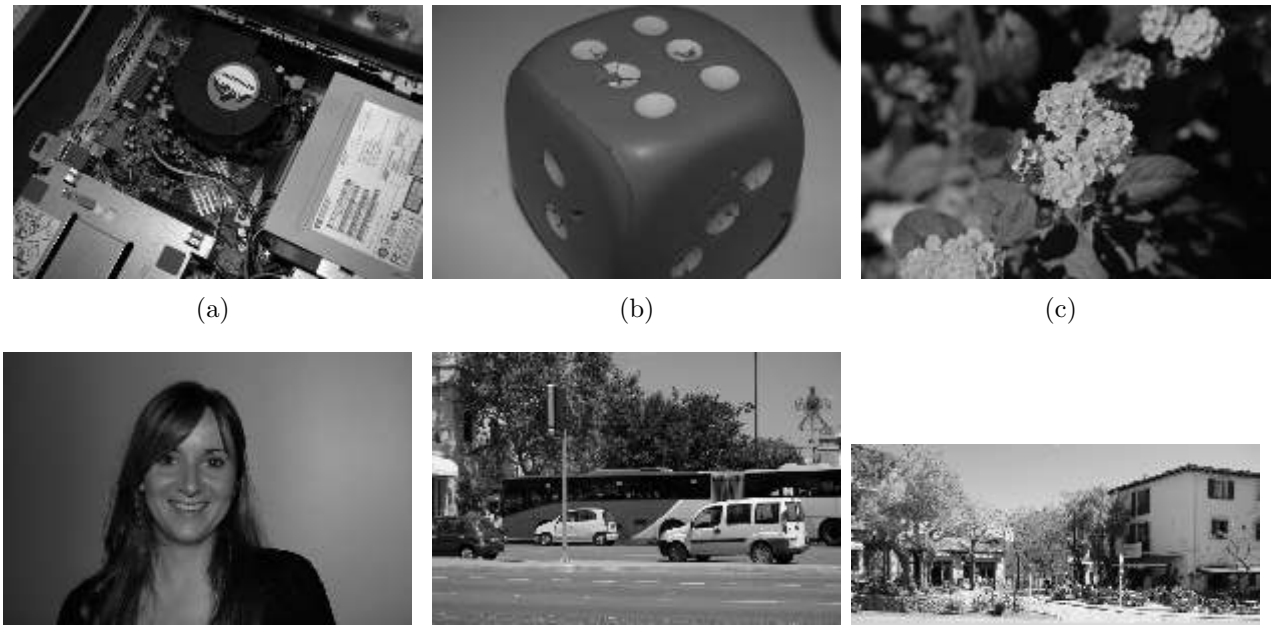
$$\mathcal{M} : \tilde{P} \in \mathcal{P} \mapsto \operatorname*{argmax}_{0 \leq k \leq 19} \mathbb{P}_{\Theta_t}(s_P = k \mid \tilde{P}).$$

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.

that more iterations on the first transformed channel generally bring about better results in terms of MSE.

Table 1 compares S-PLE with several other acclaimed algorithms [14, 15, 16, 4, 26, 18] also available on IPOL. 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.

Concluding remarks: figure 5 displays the images used in the algorithm comparison. Figure 6 makes clear that for portraits, BM3D cannot produce a relatively flat background. It is this observation that leads to the inclusion of the flat model in the mixture in the first place. And S-PLE's approach to identifying flat areas is less brutal than that of NLBayes, which explains its being able to preserve more details than NLBayes. This is amply illustrated for example on the building facade in figure 7. However, the same example also shows that that because of its intrinsically local approach to orientation detection, structure only observable on a larger scale cannot be kept in a really satisfactory manner. In addition, relative to BM3D and NLBayes, a tendency for S-PLE to overlook structural information amid strong noise is also demonstrated by the image *computer* in table 1. But given its generally superior performance and relatively low computational cost, S-PLE clearly should be counted as another state-of-the-art image denoising algorithm.



(a) (b) (c)

(d) (e) (f)

Figure 5: gray images used in the 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)

A About the Filters

In this section, we briefly explain the two types of filters deployed in the algorithm. Suppose that a random signal $S \sim \mathcal{N}(\mu_S, \Sigma_S)$ is corrupted by some Gaussian noise following $\mathcal{N}(0, \sigma^2 I)$. Wiener

Table 1: Algorithm Comparison[1]

[3]$\sigma = 2$	**PLE**[5]	**DCT**	**GSM**[4]	**KSVD**	**NLM**	**EPLL**[6]	**S-PLE**[2]	**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**

[1] The algorithms are ordered to reflect their global performance. Marked 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 [17]

[5] PLE, with no observable convergence available, iterated four times

[6] EPLL [28]

Table 2: Algorithm Comparison (Continuation)[1]

[3]$\sigma = 30$	**PLE**[5]	**DCT**	**GSM**[4]	**KSVD**	**NLM**	**EPLL**[6]	**S-PLE**[2]	**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. Marked in bold is the lowest RMSE in each row.

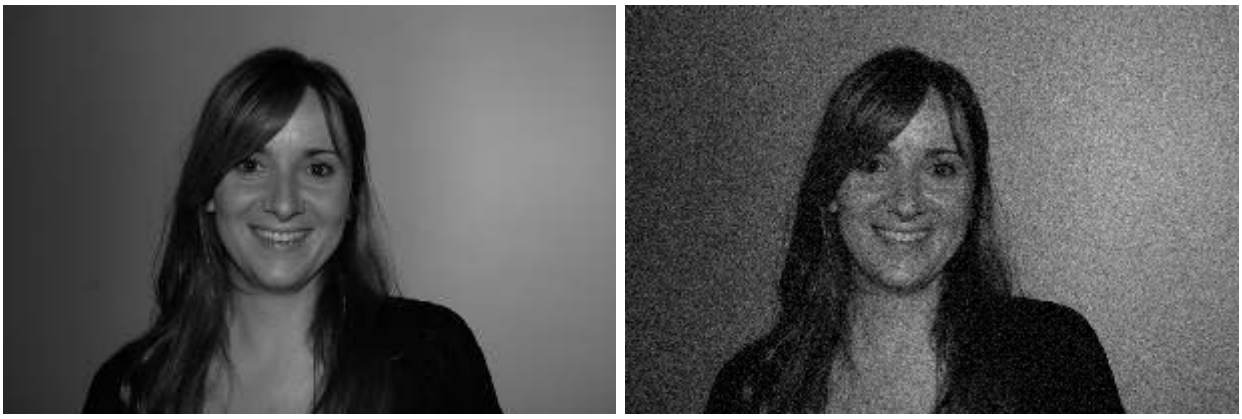
[2] S-PLE was allowed to iterate 50 times.

[3] noise standard deviation

[4] BLS-GSM [17]

[5] PLE, with no observable convergence available, iterated four times

[6] EPLL [28]



(a) (b)

(c) (d)

(e) (f)

Figure 6: (a) original *girl* 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

(a) (b)

(c) (d)

(e) (f)

Figure 7: (a) original *valldemossa* 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

filtering minimizes

$$\min_{L,b} \mathbb{E}\|L\tilde{S} + b - S\|_2^2 = \min_{L,b} \mathbb{E}\|L(S+N) + b - S\|_2^2$$

which yields the optimal linear estimate (with Σ_S's eigenvalues denoted by $(\lambda_1, \cdots, \lambda_n)$)

$$\widehat{S} = \Sigma_S(\Sigma_S + \sigma^2 I)^{-1}(\tilde{S} - \mu_S) + \mu_S = \sum_{i=1}^{n} \frac{\lambda_i}{\lambda_i + \sigma^2} \langle \tilde{S} - \mu_S, b_i \rangle b_i + \mu_S.$$

On the other hand, shrinkage uses another thresholding strategy with $\gamma_t(\cdot)$ given in definition 1

$$\widehat{S} = \sum_{i=1}^{m} \gamma_t(\langle \tilde{S} - \mu_S, b_i \rangle) b_i + \mu_S \quad \text{with } t = \sigma\sqrt{2\ln m}$$

where $(b_i)_{1 \le i \le n}$ denotes a certain basis and m satisfies $m \le n$ [9, 8].

Acknowledgements

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).

Image Credits

 by A. Buades, CC-BY

 by M. Colom, CC-BY

References

[1] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: Design of dictionaries for sparse representation. *Proceedings of Signal Processing with Adaptive Sparse Structured Representations (SPARS)*, 5:9–12, 2005.

[2] P.J. Brockwell and R.A. 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 and Simulation*, 4(2):490–530, 2006. http://dx.doi.org/10.1137/040616024.

[4] A. Buades, B. Coll, and J.M. Morel. Non-local means denoising. *Image Processing On Line*, 2011. http://dx.doi.org/10.5201/ipol.2011.bcm_nlm.

[5] P. Chatterjee and P. Milanfar. Clustering-based denoising with locally learned dictionaries. *IEEE Transactions on Image Processing*, 18(7):1438–1451, 2009. http://dx.doi.org/10.1109/TIP.2009.2018575.

[6] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image restoration by sparse 3d transform-domain collaborative filtering. In *Proceedings of SPIE 6812, Image Processing: Algorithms and Systems VI*, pages 681207–681207–12. International Society for Optics and Photonics, 2008. http://dx.doi.org/10.1117/12.766355.

[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)*, pages 1–38, 1977.

[8] D.L. Donoho and I.M. Johnstone. Adapting to unknown smoothness via wavelet shrinkage. *Journal of the american statistical association*, 90(432):1200–1224, 1995. http://dx.doi.org/10.2307/2291512.

[9] D.L. Donoho and J.M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455, 1994. http://dx.doi.org/10.2307/2337118.

[10] A.A. Efros and T.K. Leung. Texture synthesis by non-parametric sampling. In *International Conference on Computer Vision*, volume 2, pages 1033–1038. Corfu, Greece, 1999. http://dx.doi.org/10.1109/ICCV.1999.790383.

[11] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 15(12):3736–3745, 2006. http://dx.doi.org/10.1109/TIP.2006.881969.

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

[13] A.O. Hero and J.A. Fessler. Convergence in norm for alternating expectation-maximization (em) type algorithms. *Statistica Sinica*, 5(1):41–54, 1995.

[14] M. Lebrun. An Analysis and Implementation of the BM3D Image Denoising Method. *Image Processing On Line*, 2012. http://dx.doi.org/10.5201/ipol.2012.l-bm3d.

[15] M. Lebrun and A. Leclaire. An Implementation and Detailed Analysis of the K-SVD Image Denoising Algorithm. *Image Processing On Line*, 2012. http://dx.doi.org/10.5201/ipol.2012.llm-ksvd.

[16] Marc Lebrun, Antoni Buades, and Jean-Michel Morel. Implementation of the "Non-Local Bayes" (NL-Bayes) Image Denoising Algorithm. *Image Processing On Line*, pages 1–42, 2013. http://dx.doi.org/10.5201/ipol.2013.16.

[17] J. Portilla, V. Strela, M.J. Wainwright, and E.P. Simoncelli. Image denoising using scale mixtures of Gaussians in the wavelet domain. *IEEE Transactions on Image Processing*, 12(11):1338–1351, 2003. http://dx.doi.org/10.1109/TIP.2003.818640.

[18] B. Rajaei. An Analysis and Improvement of the BLS-GSM Denoising Method. *Image Processing On Line*, 2013. submitted.

[19] S. Roweis. EM algorithms for PCA and SPCA. *Advances in neural information processing systems*, pages 626–632, 1998.

[20] L.I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1-4):259–268, 1992. http://dx.doi.org/10.1016/0167-2789(92)90242-F.

[21] L. Shapiro and G.C. Stockman. Computer vision, 2001.

[22] C.M. Stein. Estimation of the mean of a multivariate normal distribution. *The annals of Statistics*, pages 1135–1151, 1981.

[23] M.E. Tipping and C.M. Bishop. Mixtures of probabilistic principal component analyzers. *Neural computation*, 11(2):443–482, 1999. http://dx.doi.org/10.1162/089976699300016728.

[24] M.E. Tipping and C.M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999. http://dx.doi.org/10.1111/1467-9868.00196.

[25] Y.Q. Wang and J.M. Morel. SURE Guided Gaussian Mixture Image Denoising. *SIAM Journal on Imaging Sciences*, 6:999–1034, 2013. http://dx.doi.org/10.1137/120901131.

[26] G. Yu and G. Sapiro. DCT image denoising: a simple and effective image denoising algorithm. *Image Processing On Line*, 2011. http://dx.doi.org/10.5201/ipol.2011.ys-dct.

[27] G. Yu, G. Sapiro, and S. Mallat. Solving inverse problems with piecewise linear estimators: from Gaussian mixture models to structured sparsity. *IEEE Transactions on Image Processing*, 21(5):2481–2499, 2012. http://dx.doi.org/10.1109/TIP.2011.2176743.

[28] D. Zoran and Y. Weiss. From learning models of natural image patches to whole image restoration. In *IEEE International Conference on Computer Vision (ICCV)*, pages 479–486. IEEE, 2011. http://dx.doi.org/10.1109/ICCV.2011.6126278.