Gaussian Priors for Image denoising

Julie Delon, Antoine Houdard

Abstract

This chapter is dedicated to the study of Gaussian priors for patch-based image denoising. In the last twelve years, patch priors have been widely used for image restoration. In a Bayesian framework, such priors on patches can be used for instance to estimate a clean patch from its noisy version, via classical estimators such as the conditional expectation or the maximum a posteriori. As we will recall, in the case of Gaussian white noise, simply assuming Gaussian (or Mixture of Gaussians) priors on patches leads to very simple closed-form expressions for some of these estimators. Nevertheless, the convenience of such models should not prevail over their relevance. For this reason, we also discuss how these models represent patches and what kind of information they encode. The end of the chapter focuses on the different ways in which these models can be learned on real data. This stage is particularly challenging because of the curse of dimensionality. Through these different questions, we compare and connect several denoising methods using this framework.

1 Introduction

This chapter focuses on patch priors for image denoising. In the last decade, patch-based models (also known as Non-local models) have created a new paradigm in image processing, leading to very significant improvements both for classical image restoration problems (denoising, *inpainting*, interpolation) or for image synthesis and editing. These models represent images by a set of local neighborhoods or *patches*, and make them collaborate regardless of their spatial position in the image, relying on the observation that most natural images present a remarkable redundancy at a semi-local scale. A patch $y_i(v)$ is a piece (most of the time a square) of an image v centered at the pixel i. As pointed out by Mumford and Desolneux [15], patches are *"the analogs of the phonemes of speech"*.

Patch-based models have been the subject of numerous works, especially in the context of image denoising. Assuming that the noise is additive, image denoising amounts to estimate an image u from its noisy version $v \in \mathbb{R}^m$ (m is the image size) such that

$$v = u + \varepsilon, \tag{1}$$

with ε a noise with known statistics (not necessarily Gaussian). In digital cameras, the two major sources of noise during the acquisition process are the thermal agitation, which produces an almost white and Gaussian noise, and the discrete nature of light, which is behind the photon *shot noise*, modeled as a Poisson variable (for a complete description of the sources of noise in a digital camera, see [2]). Stabilizing the noise variance by a generalized Anscombe transform [13] results in a noise model well approximated by a white Gaussian noise $\varepsilon \sim \mathcal{N}(0, \sigma^2 \mathrm{I}_m)$. The vast majority of works on image denoising focus on this simplified model and it is also our assumption in this chapter.

In this framework, patch-based methods usually attempts at rewriting (1) into a degradation model that can be expressed for each patch separately. All patches $\{y_i, i = 1 \dots, m\}$ of size $p = s \times s$ are first extracted

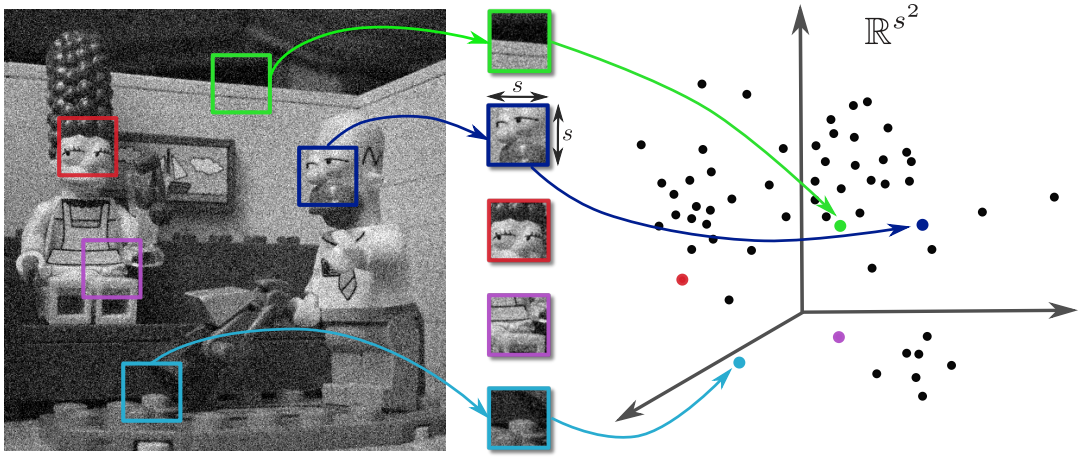

Figure 1: Image patches can be seen as vectors in a high-dimensional space. Most of the patch-based methods uses the patch-space of an image which is the set of all the sliding patches of size $p = s \times s$ extracted from the image.

from the image v and seen as noisy vectors in a high dimensional space, as illustrated by Figure 1 (in the whole chapter, when writing patches as vectors, we assume that the patches are read column-wise). Then the noisy patches are restored sequentially, before reconstructing the whole image. The degradation model on the patches becomes

$$y_i = x_i + \varepsilon_i, \;\; i \in \{1, \dots, m\} \tag{2}$$

where x_i is the patch centered at pixel i in u, y_i the same patch in v, and ε_i the additive noise. In practice, it is almost always assumed that the $\{\varepsilon_i, i = 1 \dots, m\}$ are independent samples from the Gaussian distribution $\mathcal{N}(0, \sigma^2 \mathrm{I}_p)$, although this hypothesis is obviously wrong since patches are overlapping. We will briefly discuss this issue in Section 4, along with the aggregation of the restored patches to reconstruct the whole image.

The first denoising methods relying on patches appear in 2004 [16, 21, 3, 5]. Among these methods, one of the most popular remains the Non-Local Means [5], which sees similar patches as independent realizations of the same distribution and averages these repeated structures to reduce noise variance. If numerous approaches have built on the same core ideas since 2004, the recent and most convincing approaches in patch-based denoising rely on a Bayesian reformulation of the denoising problem, using local or global statistical priors for the distribution of each patch [12, 24, 23, 20, 1, 11]. Under the white Gaussian noise model (2), the conditional distribution of a noisy patch y knowing its original version x (we omit the index i in the following) can be written

$$p(y|x) \propto e^{-\frac{\|x-y\|^2}{2\sigma^2}}. \tag{3}$$

The Bayesian model assumes that the original patch x is a realization of a random vector X with a probability distribution $p(x)$ called the *prior distribution*. Therefore, the noisy patch y is a realization of the random vector

$$Y = X + N, \tag{4}$$

with $N \sim \mathcal{N}(0, \sigma^2 I_p)$. Under these hypotheses, and assuming that N and X are independent, we can compute

the *posterior distribution*

$$p(x|y) \propto p(y|x)p(x) \propto e^{-\frac{\|x-y\|^2}{2\sigma^2}} p(x). \tag{5}$$

Ideally, in order to reconstruct the (unknown) original patch x from the degraded version y, we would like to compute the conditional expectation $\mathbb{E}[X|Y]$ (*i.e.* the mean of the posterior distribution), which minimizes the quadratic risk under the previous model. This estimator is also called the minimum mean square error (MMSE) estimator. In practice, computing this conditional expectation is often complex, and it is classical to compute instead the affine function (called linear MMSE) of Y minimizing the quadratic risk, *i.e.* the affine estimator $DY + \alpha$ (with D a $p \times p$ real matrix and α a vector in $\mathbb{R}^p$) minimizing the risk

$$\mathbb{E}[\|DY + \alpha - X\|^2].$$

This affine estimator, is called the *Wiener estimator* and will be denoted $\mathbb{E}_{Wiener}[X|Y]$ in the following. It can be easily shown by deriving the previous risk that (assuming that the following quantities exist),

$$\mathbb{E}_{Wiener}[X|Y] = \mathbb{E}[X] + \Sigma_{X,Y}\Sigma_Y^{-1}(Y - \mathbb{E}[Y]), \tag{6}$$

where $\Sigma_{X,Y} := \mathbb{E}\left[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])^t\right]$ and $\Sigma_Y := \mathbb{E}\left[(Y - \mathbb{E}[Y])(Y - \mathbb{E}[Y])^t\right]$. This affine estimator only relies on second-order moments of the signal and noise. Under model (4) and assuming that N and X are independent, the Wiener estimator becomes

$$\mathbb{E}_{Wiener}[X|Y] = \mathbb{E}[X] + \Sigma_X(\Sigma_X + \sigma^2 I_p)^{-1}(Y - \mathbb{E}[Y]), \tag{7}$$

with Σ_X the covariance matrix of the random vector X.

Another classical solution to reconstruct x is to compute the maximum (MAP) of the *a posteriori* distribution $p(y|x)$, which yields:

$$\begin{aligned} \widehat{x}(y) = \arg\max_{x \in \mathbb{R}^p} p(x|y) &= \arg\max_{x \in \mathbb{R}^p} p[y|x]\ p(x) \\ &= \arg\min_{x \in \mathbb{R}^p} -\log p(y|x) - \log p(x) \\ &= \arg\min_{x \in \mathbb{R}^p} \frac{\|x-y\|^2}{2\sigma^2} - \log p(x). \end{aligned}$$

From this point of view, restoring each patch is equivalent to solve a variationnal problem, with a quadratic fidelity term and a smoothness term derived from the prior.

The most convenient prior for computing the previous estimators is the Gaussian distribution. Indeed, on the one hand, Gaussian priors are well suited to encode patch structures with some kind of contrast invariance, as we will see in Section 2. On the other hand, under a Gaussian prior, the conditional expectation, Wiener estimator and MAP coincide, as we will see in Section 3. For these reasons, these priors are favored in most recent works on patch-based image denoising [6, 12, 1]. A slightly more involved prior used in the literature is the Gaussian Mixture Model (GMM) [24, 19, 23, 20, 11]. In this case, computing the conditional expectation remains simply tractable. All these works differ among other things in the way they infer the parameters of the Gaussian or GMM distributions. These distributions live in $\mathbb{R}^p$ and estimation in such high-dimensional spaces is complex. We will see in Section 5 the different possibilities to infer these parameters and how some of these works tackle the curse of dimensionality. Figure 2 illustrates the main steps common to all these patch based denoising methods, and each of these steps is described in the following sections.

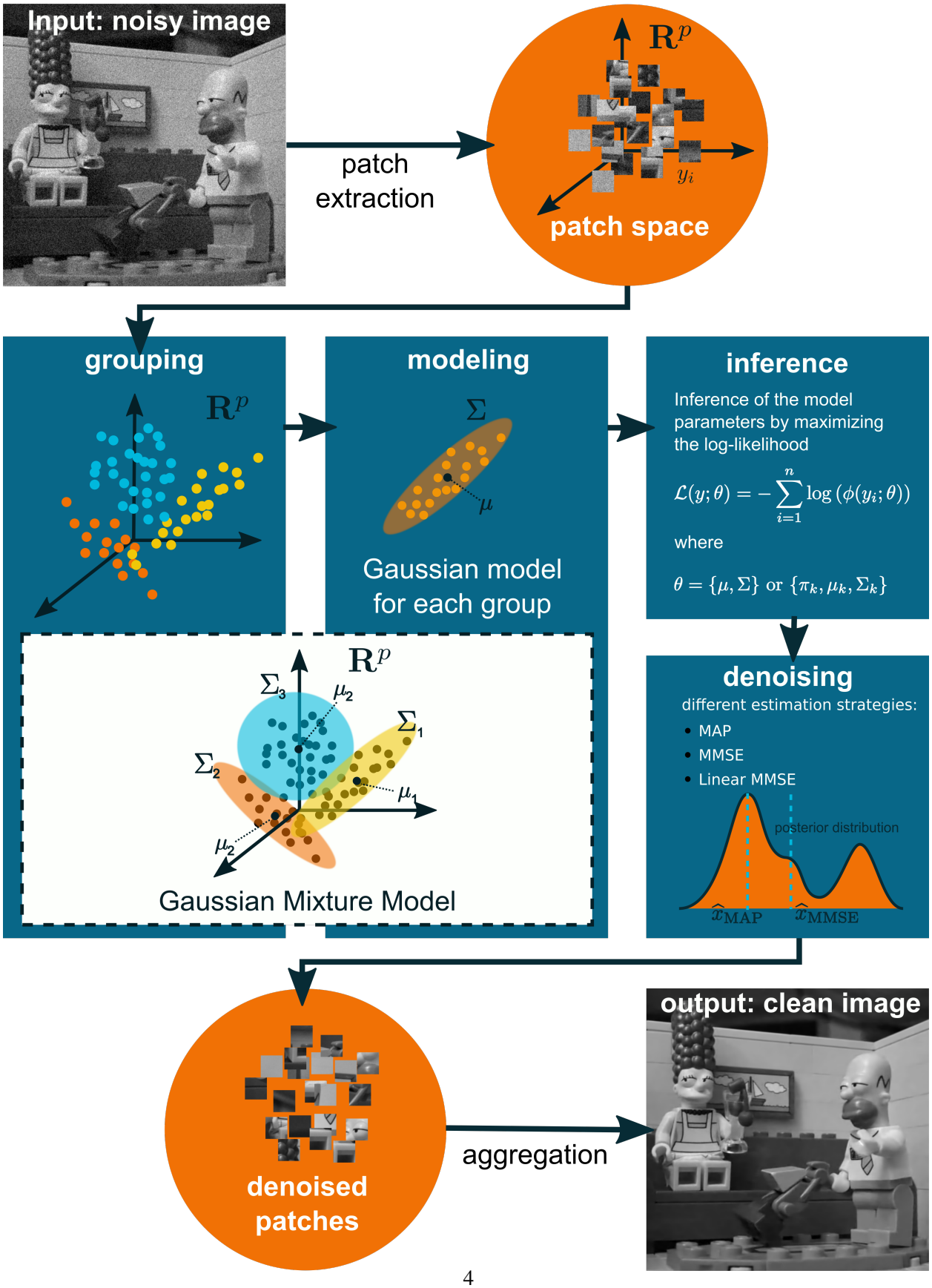

Figure 2: The whole process of patch-based image denoising with Gaussian prior models. First, patches are extracted from the noisy image. Next, these noisy patches are grouped and modeled with local Gaussians or Gaussian Mixture Models, whose parameters are inferred by maximum likelihood (Section 5). Each patch is then denoised with an estimator derived from the model (Section 3). Finally, the clean patches are aggregated to recover the denoised image (Section 4).

2 What is encoded in Gaussian and GMM priors ?

Before going into the details of estimation under Gaussian priors, we provide in this section a few insights on the actual structures they encode. Assume a Gaussian model $\mathcal{N}(\mu,\Sigma)$ for $p = s \times s$ patches ($\mu \in \mathbb{R}^p$ and $\Sigma \in \mathscr{M}_p(\mathbb{R})$). The diagonal coefficients of the covariance matrix Σ represent the variance of each pixel in the patch, while the non-diagonal coefficients represent the covariances between pixels. A positive covariance coefficient means that the two pixels tend to be either both greater or smaller than their means, while a negative coefficient implies that they tend to be on opposite sides of their means. Clearly, if Σ is purely diagonal, patches drawn from the model $\mathcal{N}(\mu,\Sigma)$ will only be noisy versions of the mean patch μ. In this case, the only structure information is contained in μ. More interesting models contain geometric information directly in the covariance matrix Σ.

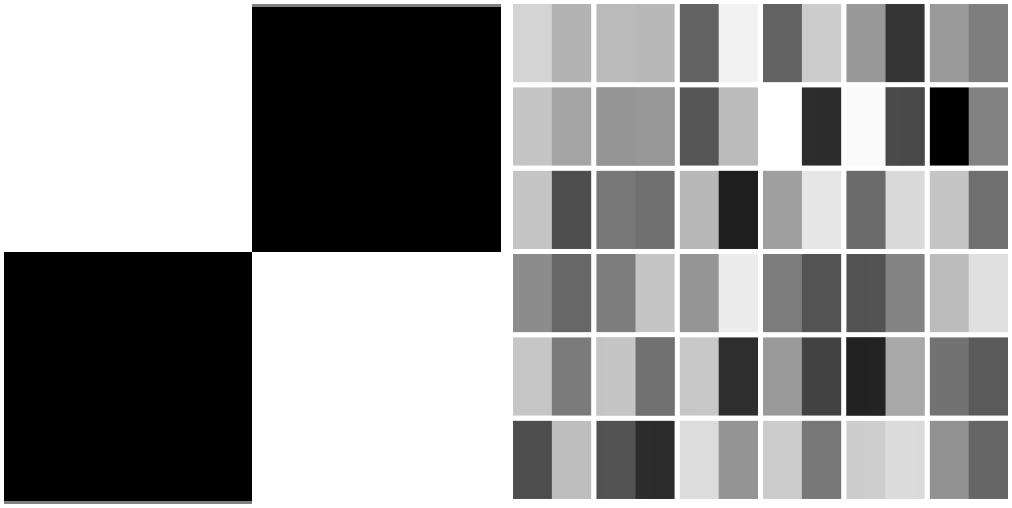

Figure 3: Left: a covariance matrix Σ with 1 (white) on the second and third quarters, and 0 (black) on the first and fourth quarters. Right: patches drawn from the Gaussian distribution $\mathcal{N}(\mu,\Sigma)$ with μ a constant patch equal to 0.5.

To illustrate this point, we propose to create models encoding different patch structures. For instance, in order to model a vertical edge, we define a Gaussian distribution with constant mean $\mu = (0.5,\dots,0.5)$ and a covariance matrix with coefficient 1 in the second and third quarter of Σ, and coefficient 0 in the first and fourth quarters of Σ (see Figure 3). In this simplistic example, the matrix Σ has rank two, with (non trivial) eigenvectors $(1,\dots,1,0,\dots,0)$ and $(0,\dots,0,1,\dots,1)$, so all the patches drawn from this distribution can be written $0.5+(\alpha,\dots,\alpha,\beta,\dots,\beta)$ with $\alpha \sim \mathcal{N}(0,1)$ and $\beta \sim \mathcal{N}(0,1)$. These patches all contain a vertical edge in their middle, with grey levels α and β on both sides of the edge. In this example, we see that the model encodes a structure and authorizes different contrasts on both sides of the structure. With the same mechanic, we can create a covariance matrix encoding any desired shape, see for instance Figure 4. Again, the samples from the corresponding distribution exhibit all possible grey levels in the different regions defined by the covariance matrix, even if all these grey levels are not all equally likely.

Now, although these models authorize contrast changes or contrast inversions, they are not well suited to encode geometric invariances on patches. For instance, if we try to learn a model encoding different vertical edges with invariance to translation, we end up with an average model encoding a vertical gradient image (see Figure 5).

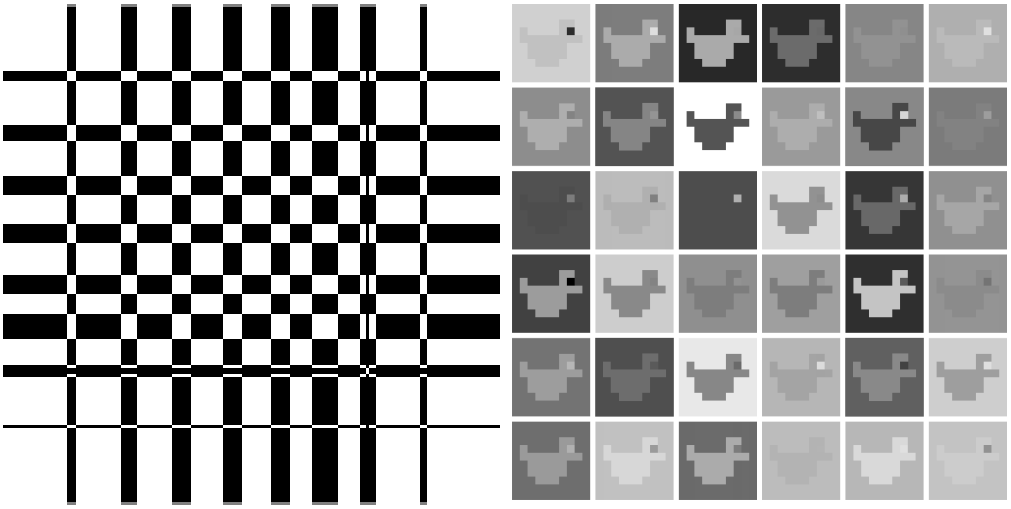

Figure 4: Left: a covariance matrix Σ composed of 1 (white) and 0 (black). Right: patches drawn from the Gaussian distribution $\mathcal{N}(\mu,\Sigma)$ with μ a constant patch equal to 0.5.

3 How to derive estimators under Gaussian and GMM priors

Now that we have seen more precisely what could be contained in Gaussian priors, we will now see more precisely how they can be used to derive estimators under the Bayesian model described in the introduction.

In the whole section, we assume that we work with the model (4)

$$Y = X + N,$$

with $N \sim \mathcal{N}(0,\sigma^2 I_p)$ independent from X. We wish to estimate X knowing Y.

We first recall some classical results on the conditioning of Gaussian vectors, and on the links between the conditional expectation, Wiener estimator and MAP for Gaussian and GMM priors. These different estimators will serve in the rest of the chapter as denoising strategies for image patches.

3.1 Estimation with Gaussian priors

We first assume that X follows a Gaussian distribution $\mathcal{N}(\mu_X,\Sigma_X)$ and that the noise N is independent from X. The classical properties of Gaussian vectors make it possible to show that in this case the estimator $\mathbb{E}[X|Y]$ is an affine function of Y (thus equivalent in this case to the Wiener estimator). Indeed, recall that if (T,V) is a Gaussian vector, then the conditional expectation $\mathbb{E}[T|V]$ is the affine function of V

$$\mathbb{E}[T|V] = \mathbb{E}[T] + \Sigma_{T,V}\Sigma_V^{-1}(V - \mathbb{E}[V]), \tag{8}$$

where Σ_V is the covariance matrix of V and $\Sigma_{T,V} = \mathbb{E}[(T-\mathbb{E}[T])(V-\mathbb{E}[V])^t]$ (if Σ_V is not full rank, the result is still true by taking the Moore-Penrose pseudo-inverse of Σ_V).

Now, if X and N are independent Gaussian random vectors, the concatenated vector $(X,Y) = (X, X+N)$ is also Gaussian. We directly deduce the following result.

Proposition 1. *Assume that X and Y follow the model* (4)*, with $X \sim \mathcal{N}(\mu_X,\Sigma_X)$ and $N \sim \mathcal{N}(0,\sigma^2 I_p)$ independent, then the conditional expectation and Wiener estimator of X knowing Y coincide and can be written*

$$\mathbb{E}[X|Y] = \mathbb{E}_{Wiener}[X|Y] = \mu_X + \Sigma_X(\Sigma_X + \sigma^2 I_p)^{-1}(Y - \mu_X).$$

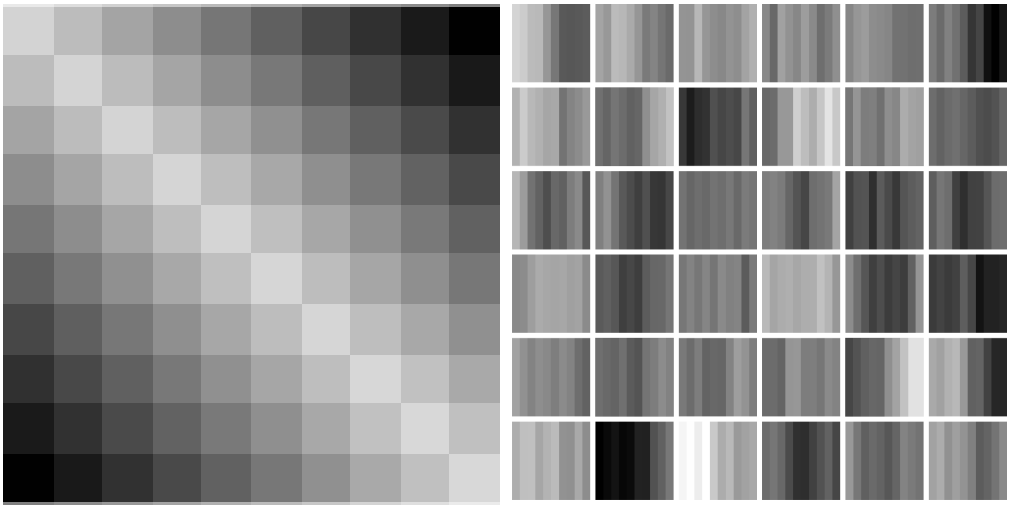

Figure 5: Left: a covariance matrix Σ learned as the sample covariance matric of a set of vertical edges at different spatial positions, and with also different choices of grey levels on both sides of the edge. Right: patches drawn from the corresponding Gaussian distribution $\mathcal{N}(\mu,\Sigma)$ with μ a constant patch equal to 0.5.

Proof. On the one hand, since (X,Y) is a Gaussian vector, the conditional expectation $\mathbb{E}[X|Y]$ can be written

$$\begin{aligned}\mathbb{E}[X|Y] &= \mathbb{E}[X]+\Sigma_{X,Y}\Sigma_Y^{-1}(Y-\mathbb{E}[Y]) \\ &= \mathbb{E}[X]+\mathbb{E}[(X-\mathbb{E}[X])(X+N-\mathbb{E}[X+N])^t](\Sigma_X+\sigma^2 I_p)^{-1}(Y-\mathbb{E}[Y]). \\ &= \mathbb{E}[X]+\Sigma_X(\Sigma_X+\sigma^2 I_p)^{-1}(Y-\mathbb{E}[Y]) = \mu_X+\Sigma_X(\Sigma_X+\sigma^2 I_p)^{-1}(Y-\mu_X).\end{aligned}$$

□

Under the same hypothesis, if we try to maximize the *a posteriori* probability on the patch X, we obtain

$$\begin{aligned}\arg\max_X \log\mathbb{P}[X|Y] &= \arg\max_X \ (\log\mathbb{P}[Y|X]+\log\mathbb{P}[X]) \\ &= \arg\min_X \ \left((X-Y)^t(X-Y)/\sigma^2+(X-\mathbb{E}[X])^t\Sigma_X^{-1}(X-\mathbb{E}[X])\right).\end{aligned}$$

We check easily that the solution of this minimization problem is also given by

$$\psi(Y)=\mu_X+\Sigma_X(\Sigma_X+\sigma^2 I_p)^{-1}(Y-\mu_X).$$

Said otherwise, for a Gaussian prior, the MMSE, linear MMSE and MAP all coincide and all these estimators only require linear operations. This property makes Gaussian priors particularly convenient in practice and explains their success in the restoration literature.

We can illustrate the interest of this estimator on the Gaussian model $\mathcal{N}(\mu_X,\Sigma_X)$ presented on Figure 3 and representing a vertical edge. If X is an (unknown) realization of this model and $Y=X+N$ with $N\sim\mathcal{N}(0,\sigma^2 I_p)$ independent from X, then $\mathbb{E}[X|Y]$ will also be a patch $(\alpha,\ldots,\alpha,\beta,\ldots,\beta)$ with $\alpha=0.5+\frac{1}{p/2+\sigma^2}\sum_{k=1}^{p/2}(Y_k-0.5)$ and $\beta=0.5+\frac{1}{p/2+\sigma^2}\sum_{k=p/2+1}^{p}(Y_k-0.5)$(assuming p is even for the sake

of simplicity). Said otherwise, the denoised patch $\mathbb{E}[X|Y]$ represents the same vertical edge as X and its values α and β on both sides of the edge are (if $\sigma^2 << p/2$) the averages of Y on these two half patches.

Figure 6 represents three denoising experiments with the previous estimator. On the first line, a vertical edge is denoised with the Gaussian model of Figure 3. On the second line, a "duck" patch is denoised with the Gaussian model of Figure 4. In both cases, using the conditional expectation works extremely well because the Gaussian model used in the estimator fits perfectly the image to be denoised. On the third line, the noisy edge is denoised with the Gaussian model of Figure 5. In this case, the denoised patch is constant on each column (since the model is learned from a set of translated vertical edges). Although the model imposes a strong correlation between columns of the first half of the patch on the one hand, and between columns of the second half of the patch on the other hand, this is not enough to restore the patch perfectly.

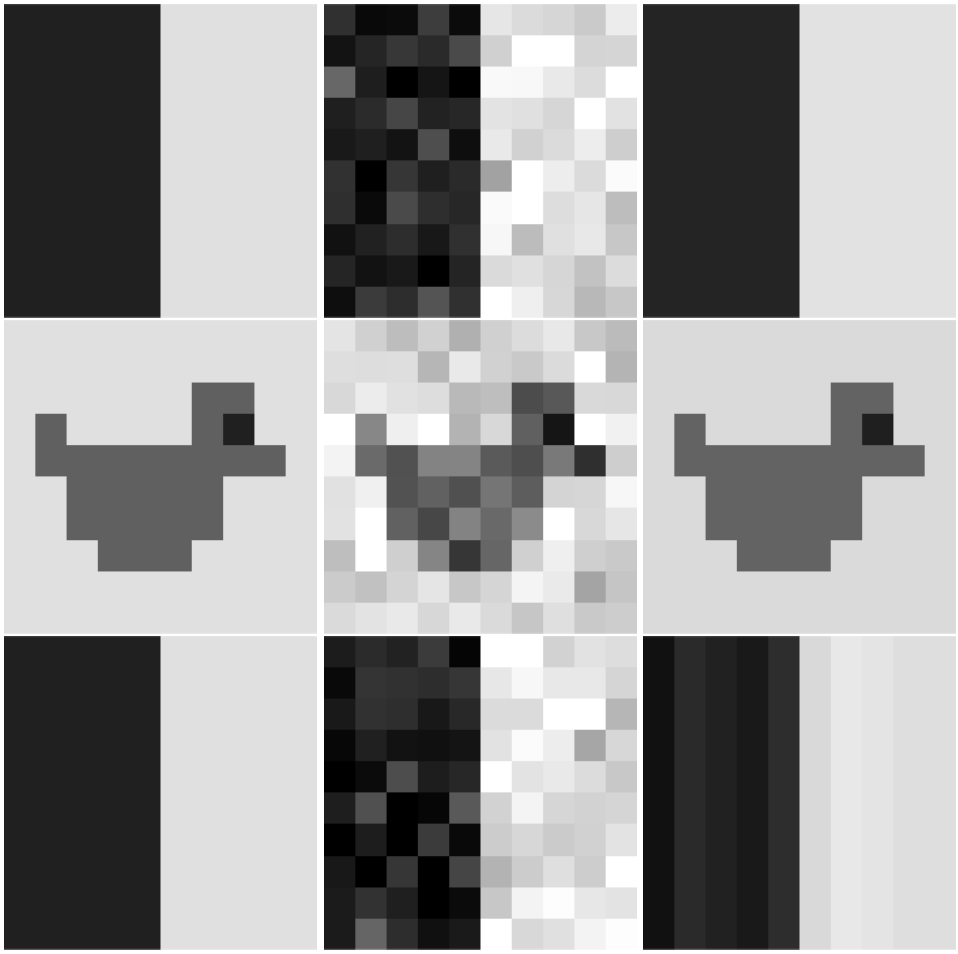

Figure 6: For each line, from left to right, clean patch, noisy patch ($\sigma = 10\%$), denoised patch with the Wiener estimator. First line, the edge Gaussian model of Figure 3 is used to denoise (PSNR = 37.17). Second line, the duck Gaussian model of Figure 4 is used to denoise (PSNR = 34.29). Third line, the gradient model of Figure 5 is used to denoise (PSNR = 29.68). In this last case, the image to be denoised is not well represented by the model and the result is less convincing.

3.2 Estimation with Gaussian Mixture Models

The case of Gaussian Mixture Models is a bit more involved but remains globally simple. Assume that X follows a Gaussian Mixture Model

$$X \sim \sum_{k=1}^{K} \pi_k \mathcal{N}(\mu_k, \Sigma_k), \tag{9}$$

with $\sum_{k=1}^{K} \pi_k = 1$. There exists a latent random variable Z on $\{1,\dots,K\}$ such that $\mathbb{P}[Z=k] = \pi_k$ and such that $X|Z=k \sim N(\mu_k, \Sigma_k)$. In the following, we note $\psi_k(y)$ the Wiener estimator for the k^{th} Gaussian, *i.e.*

$$\psi_k(y) = \mu_k + \Sigma_k(\Sigma_k + \sigma^2 I_p)^{-1}(y - \mu_k).$$

Under this model, we have the following proposition.

Proposition 2. *Assume that X and Y follow the model* (4)*, with $X \sim \sum_{k=1}^{K} \pi_k \mathcal{N}(\mu_k, \Sigma_k)$ and $N \sim \mathcal{N}(0, \sigma^2 I_p)$ independent, then the conditional expectation of X knowing Y can be written*

$$\mathbb{E}[X|Y] = \sum_{k=1}^{K} \psi_k(Y)\mathbb{P}[Z=k|Y]. \tag{10}$$

Proof. To compute the conditional expectation, we can start by noting that if $Z=k$, (X,Y) is a Gaussian vector and the results of the previous section apply. We can now compute the conditional expectation

$$\mathbb{E}[X \mid Y, Z] = \psi_Z(Y) = \sum_{k=1}^{K} \psi_k(Y)\mathbf{1}_{Z=k}.$$

It follows that

$$\begin{aligned}
\mathbb{E}[X|Y] &= \mathbb{E}[\mathbb{E}[X \mid Y,Z] \mid Y] \quad \text{because} \quad \sigma(Y) \subset \sigma(Y,Z) \\
&= \mathbb{E}[\psi_Z(Y) \mid Y] = \sum_{k=1}^{K} \mathbb{E}[\psi_k(Y)\mathbf{1}_{Z=k} \mid Y] \\
&= \sum_{k=1}^{K} \psi_k(Y)\mathbb{E}[\mathbf{1}_{Z=k} \mid Y] \quad \text{because} \quad \psi_k(Y) \text{ is } \sigma(Y)\text{-measurable.}
\end{aligned}$$

We deduce that

$$\mathbb{E}[X|Y] = \sum_{k=1}^{K} \psi_k(Y)\mathbb{E}[\mathbf{1}_{Z=k} \mid Y] = \sum_{k=1}^{K} \psi_k(Y)\mathbb{P}[Z=k \mid Y].$$

□

The conditional expectation $\mathbb{E}[X|Y]$ can be seen as a linear combination of affine functions of Y, with weight $\mathbb{P}[Z=k|Y]$ representing the probability that the patch belongs to the class k. However, the weights $\mathbb{P}[Z=k \mid Y]$ are not linear functions of Y.

The expression of the Wiener estimator $\mathbb{E}_{Wiener}[X|Y]$ can be deduced directly from Equation (7), by replacing $\mathbb{E}[X]$ by $\sum_{k=1}^{K} \pi_k \mu_k$ and Σ_X by the complete covariance of the GMM.

Finally, computing the MAP $\arg\max_X \log \mathbb{P}[X|Y]$ under a GMM prior on X is much less convenient and does not lead to a closed-form solution. Indeed, it boils down to compute the maximum of the posterior distribution, which is another Gaussian Mixture distribution.

In other words, the linear MMSE, MMSE and MAP do not coincide for Gaussian Mixture priors. In practice, the conditional expectation is favored since it is much simpler to compute than the MAP.

3.3 Other estimation strategies

Estimation under Gaussian or GMM models has several links with other estimation strategies found in the literature. For a noisy patch y, and a Gaussian model $\mathcal{N}(\mu,\Sigma)$, we have seen that the conditional expectation strategy consists in computing the denoised patch

$$\widehat{x}(y) = \mu + \Sigma(\Sigma + \sigma^2 I_p)^{-1}(y-\mu).$$

Now, if we consider the eigendecomposition $\Sigma = Q\Delta Q^t$ with $\Delta = \mathrm{diag}(\lambda_1,\ldots,\lambda_p)$, this can be rewritten

$$\widehat{x}(y) = \mu + Q\mathrm{diag}\left(\frac{\lambda_1}{\lambda_1+\sigma^2},\ldots,\frac{\lambda_p}{\lambda_p+\sigma^2}\right)Q^t(y-\mu). \tag{11}$$

More generally, denoting $Q_1,\ldots,Q_p$ the columns of Q representing the eigenvectors, we can write

$$\widehat{x}(y) = \mu + \sum_{k=1}^{p} \eta_k\left(\langle Q_k | y-\mu\rangle\right) Q_k, \tag{12}$$

with $\eta_k(z) = \frac{\lambda_k}{\lambda_k+\sigma^2}z$. Although the previous Wiener estimator is used in numerous recent patch-based denoising methods [12, 19, 20], other choices are obviously possible for η_k, such as hard or soft thresholding [8], or all estimators classically used in diagonal estimation.

Writing $\tilde{x} = Q^t(x-\mu)$, we can see that the conditional expectation $\widehat{x}(y)$ is also solution of the optimization problem

$$\underset{\tilde{x}}{\mathrm{argmin}}\, \|Q\tilde{x}-(y-\mu)\|^2 + \sigma^2\tilde{x}^t\Delta^{-1}\tilde{x} = \underset{\tilde{x}}{\mathrm{argmin}}\, \|Q\tilde{x}-(y-\mu)\|^2 + \sigma^2\sum_{k=1}^{p}\frac{\tilde{x}_j^2}{\lambda_k}.$$

This permits to see the link between the previous approach and the dictionary-based approaches, the dictionary here being given by Q and the second term corresponding to a regularization of the solution $\tilde{x}$. Figure 7 represents the denoising of a noisy patch with the same Gaussian model and two different denoising strategies: the conditional expectation (Wiener) and hard thresholding at 2.7σ (as recommended in [8]).

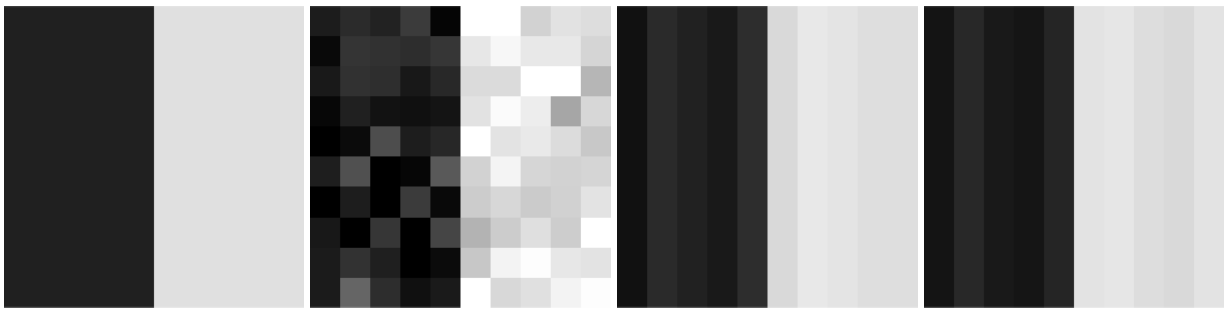

Figure 7: Clean patch, noisy patch (10% noise), denoised patch with gradient model (from Fig. 5) and Wiener estimator (PSNR = 29.68dB), and denoised patch with gradient model and hard thresholding (PSNR = 31.12dB, th = 2.7σ)

4 From patches to images: aggregation procedures

In the previous sections, we have seen how to derive bayesian estimators to perform denoising on each patch separately. In this framework, each observed patch y_i from a noisy image v is denoised into $\widehat{x_i}$, which is an estimate of the unknown patch x_i. Each pixel of the image v is contained in p patches, which provide p denoised versions for this pixel. Most aggregation procedures consists in defining a reprojection function $\psi : \mathbb{R}^{m\times p} \to \mathbb{R}^m$ which reconstructs an image from the set of its denoised patches. Observe that since denoised patches usually do not coincide on their overlap, this operation is not invertible. Moreover, since the noise on overlapping patches is not independent, the p denoised versions of the pixel carry this dependence under the form of low-frequency noise. In the literature, we find three main strategies for this reprojection step:

- **Central pixel reprojection**. The idea is to keep only the central pixel of each denoised patch.
- **Uniform reprojection.** All the estimators coming from the different patches containing the pixel are averaged with uniform weights. This strategy is the most commonly used in practice, and this is the one we use in this chapter for the sake of simplicity.
- **Weighted reprojection.** All the estimators coming from the different patches containing the pixel are averaged with weights representing the precision of the corresponding estimator. For some details see [18, 17, 6].

A more involved strategy is explored in [24]. The authors propose to reconstruct the denoised image u as the solution of

$$\operatorname*{argmin}_u \frac{\lambda}{2}\|u-v\|_2^2 - \sum_j \log\ p(x_j),$$

where the $\{x_j\}$ are the patches extracted from the unknown image u and p is a GMM prior on the image patches. This formulation includes both the denoising and aggregation step into a single variational problem.

5 Inference of Gaussian and GMM priors

Gaussian models and GMMs appear to be well suited for patch based denoising. However, the quality of the restoration strongly depends on the relevance of the model. Unfortunately, in real denoising problems the perfect model is never known and the most challenging step is to find a good prior for each patch. In the literature, we find essentially two strategies to learn these models. The first one consists in learning the model on some external set of patches that represent the diversity of natural images [24]. The second one consists in learning the model directly on the noisy patches [19, 12, 11]. In this section, we discuss different approaches adopting the second strategy. Before going further, we recall some basics about statistical inference.

Given a set of patches $\{y_1,\dots,y_n\} \in \mathbb{R}^p$ extracted from an image, we consider them as independent realizations of a random variable Y with density ϕ depending on some parameters θ. The parameters θ of the model are inferred by maximizing the likelihood of the data *w.r.t.* θ, where the likelihood is defined as

$$\ell(y;\theta) = \prod_{i=1}^{n} \phi(y_i;\theta). \tag{13}$$

Maximizing the likelihood is equivalent to minimize the negative log-likelihood

$$\mathscr{L}(y;\theta) = -\log\left(\ell(y;\theta)\right) = -\sum_{i=1}^{n} \log\left(\phi(y_i;\theta)\right), \tag{14}$$

which is usually more convenient for computation.

In the context of denoising, we put a prior model on the random vector X representing the clean patches. When X follows a Gaussian model of parameters (μ_X, Σ_X), resp. a Gaussian mixture model of parameters $\{\pi_k, \mu_k, \Sigma_k\}_{k=1\ldots K}$, then $Y = X + N$ also follows a Gaussian model of parameters $\{\mu_X, \Sigma_X + \sigma^2 I\}_k$, resp. a GMM of parameters $(\pi_k, \mu_k, \Sigma_k + \sigma^2 I)$. Since Σ_X (resp. Σ_k) is positive semi-definite and $\sigma > 0$, $\Sigma_X + \sigma^2 I$ (resp. $\Sigma_k + \sigma^2 I$) is always positive definite. Thus, the random vector Y always has a probability density function ϕ and the likelihood is always defined.

5.1 Gaussian models

In the case of a Gaussian prior $X \sim \mathcal{N}(\mu_X, \Sigma_X)$ on the clean patches, the set of parameters on the noisy patches is given by $\theta = \{\mu_Y, \Sigma_Y\}$ where $\Sigma_Y = \Sigma_X + \sigma^2 I$ and $\mu_X = \mu_Y$. The negative log-likelihood for a set of noisy data $\{y_1, \ldots, y_n\}$ becomes

$$\mathscr{L}(y; \theta) = \frac{1}{2} \sum_{i=1}^{n} (y - \mu_Y)^T {\Sigma_Y}^{-1} (y - \mu_Y). \tag{15}$$

The computation of the maximum likelihood estimators (MLE) of the parameters, *i.e.* $\operatorname{argmin}_\theta \mathscr{L}(x; \theta)$, for μ_Y and Σ_Y yields the sample mean

$$\widehat{\mu}_Y(n) = \frac{1}{n} \sum_{i=1}^{n} y_i, \tag{16}$$

and the sample covariance matrix

$$\widehat{\Sigma}_Y(n) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \widehat{\mu}_Y)^T (y_i - \widehat{\mu}_Y). \tag{17}$$

Theses estimators depend on the number n of samples and from the strong law of large numbers

$$\widehat{\mu}_Y(n) \xrightarrow[n\to\infty]{a.s.} \mu_Y \text{ and } \Sigma_Y(n) \xrightarrow[n\to\infty]{a.s.} \Sigma_Y. \tag{18}$$

This gives us an estimator $\widehat{\Sigma}_X := \widehat{\Sigma}_Y - \sigma^2 I$ for Σ_X satisfying

$$\widehat{\Sigma}_X(n) \xrightarrow[n\to\infty]{a.s.} \Sigma_X. \tag{19}$$

In summary, for a given set of noisy patches $\{y_1, \ldots, y_n\}$ we can easily compute the MLE of the parameters (μ_X, Σ_X) for the Gaussian model on the underlying clean patches. Now, since we showed in Section 2 that Gaussian models are representing really precise structures, the most challenging part is to choose the set of noisy patches from which the model can be derived.

5.2 How to group patches to infer Gaussian priors?

In this section, we discuss how patches can be grouped in order to learn the previous Gaussian models directly from a noisy image.

5.2.1 Global Gaussian prior

The first really basic idea is to model the set of all image patches with a unique Gaussian prior. In this case, we are modeling the whole "patch-space" by a unique Gaussian model of mean $\widehat{\mu}_X$ and covariance $\widehat{\Sigma}_X$. This model poorly represents the complexity of the patch-space but still encodes some proper image information. This modeling is adopted in [8] to perform a basic denoising by performing the eigendecomposition $\Sigma_X = Q\Delta Q^t$ and denoising the patches with an estimator of the form (12). Figure 8 illustrates the fact that the eigenvectors of the covariance matrix learned on the whole patch space encode some proper information about the image.

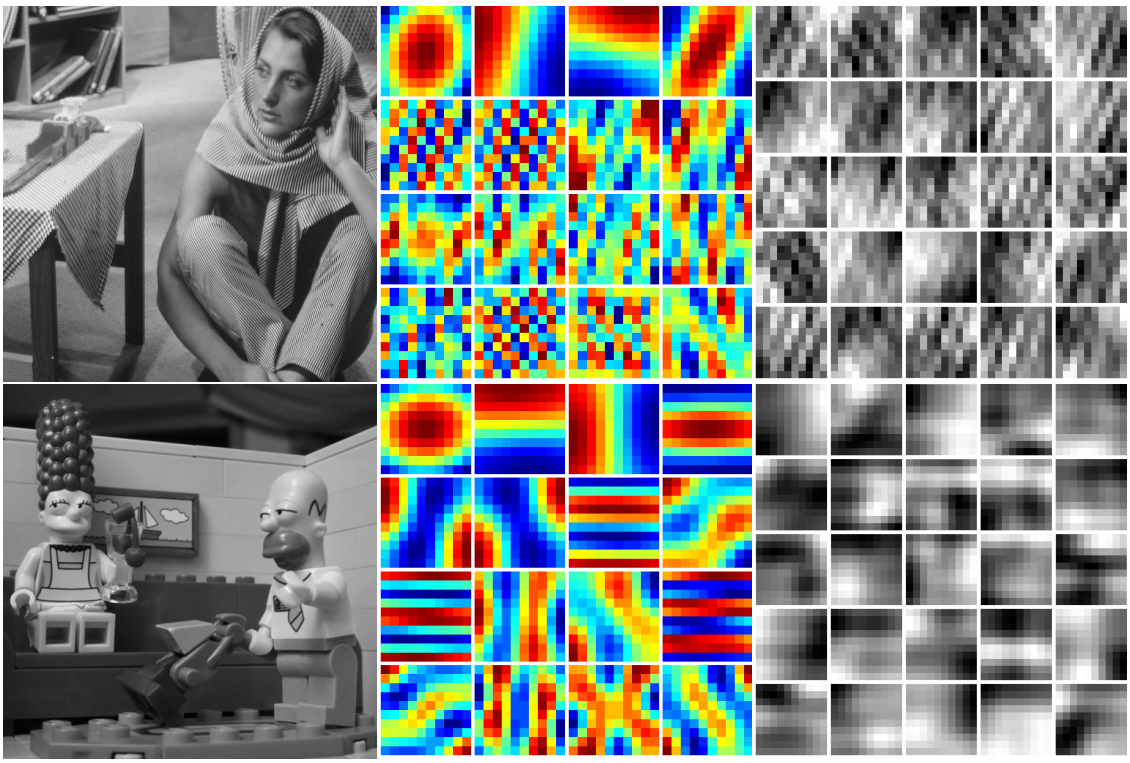

Figure 8: Visualization of the first 16 eigenvectors of the sample covariance matrix of the whole patch space for two different images. Left: original images. Middle: the 16 first eigenvectors. Right: patches generated with the low rank covariance matrix created from these eigenvectors.

In this case, since the Gaussian model is very broad, we do not expect the Wiener estimator to yield good results. But since the eigenbasis seems to encode some proper information about the image patches, the hard thresholding strategy manages surprisingly good denoising. The second line of Figure 9 shows the denoising result for this global grouping with the two denoising strategies and shows that in this case, the hard-thresholding strategy is better than the Wiener one.

5.2.2 Spatially local Gaussian priors

To derive more precise prior models, it is necessary to group "similar" patches and to restrict the inference to each of these groups. A first possibility is to group patches based on their spatial proximity in the image. This makes sense in homogeneous regions, but the risk is high to group patches representing really different

structures. The third line of Figure 9 shows that the result of this strategy is not really better, PSNR-wise, than the result of the global strategy. However, the Wiener strategy for this local approach seems nicer than in the global approach, while the result of the hard-thresholding strategy does not really change.

5.2.3 Local Gaussian priors in the space of patches

In order to learn more precise models, patches can be clustered directly in the patch space and a Gaussian model can be inferred for each cluster. All patches from the cluster can then be denoised using this model. This clustering implies to use an appropriate similarity measure between patches. The fourth line of Figure 9 shows such a denoising experiment with a K-means clustering relying on the Euclidean distance, with $K = 256$ clusters (Figure 10 shows the corresponding clustering). This usually yields a better denoising than the global and the local grouping strategies.

This way of grouping patches in the patch space together with a Wiener filtering is also one of the main ideas behind the two steps of the NL-Bayes algorithm [12]. In this algorithm, each patch y_i is associated with the group of all its ε-close patches for the Euclidean norm. A Gaussian model is inferred from this group and the whole group is denoised using this model. The final estimator for each patch is the average of all its denoised versions. The NL-Bayes algorithm uses this strategy twice: in the first step, distances are computed directly between noisy patches in $\mathbb{R}^p$; in the second step, distances between patches are computed between the versions which have denoised during the first step. Grouping ε-close patches presents the advantage of putting together patches representing the same structures. However, a straightforward one-step implementation (fifth row of Figure 9) of this idea shows that it does not work as well as expected in practice. Two major issues arise in this context:

- The high dimensionality of the patch space makes the estimation of the covariance matrix difficult;
- The use of the Euclidean distance for grouping does not allow similar patches with different contrast to be in the same group, which is a loss because we saw in Section 2 that a Gaussian model can encode information up to contrast changes.

The first issue, discussed in Section 5.4, is crucial and related to the curse of dimensionality. Unfortunately, it is hardly taken into account in the image denoising literature.

To tackle the second issue, other norms were investigated in the literature [7]. Another idea is to use the Gaussian models previously learned for recalculating new clusters. Indeed, each covariance matrix of the different Gaussian models provides a semi-norm that can be used to recompute the ε-nearest patches of each group.

5.3 Inference for Gaussian Mixture Models

The inference in the case of a mixture model is slightly more challenging since a direct maximization of the likelihood is not possible. The negative log-likelihood of the noisy data $\{y_1, \dots, y_n\}$ is given by

$$\mathscr{L}(y;\theta) = \sum_{i=1}^{n} \log\left(\sum_{k=1}^{K} \pi_k \phi(y_i;\theta_k)\right) \tag{20}$$

and the minimization of this function w.r.t θ is a complex problem. However, if we know to which group each sample x_i belongs, the log-likelihood becomes

$$\mathscr{L}(y,z;\theta) = \sum_{i=1}^{n}\sum_{k=1}^{K} z_{ik} \log\left(\pi_k \phi(y_i;\theta_k)\right) \tag{21}$$

with $z_{ik} = 1$ if y_i belongs to the group k and 0 otherwise. $\mathscr{L}(y, z; \theta)$ is the log-likelihood of the data completed with the latent random variable Z that determines the group from which the observations come from, that is $Y_i|(Z_i = k) \sim \mathcal{N}(\mu_k, \Sigma_k)$ and $p(Z_i = k) = \pi_k$.

The EM algorithm consists in iterating two steps ; the expectation (E) step that calculates the expected value of (21) with respect to the conditional distribution of Z given Y for the current value of the parameters θ. And the maximization (M) step that consists in the update of the parameters by minimizing the expectation of the complete log-likelihood from the E-step:

$$\mathbf{E}\left(\mathscr{L}(y, z; \theta)\right) = \sum_{i=1}^{n} \sum_{k=1}^{K} \mathbf{E}(z_{ik}|x_i, \theta) \log\left(\pi_k \phi(y_i; \theta_k)\right) \tag{22}$$

which leads to tractable expressions for the MLE of the parameters. It can be shown (see for example [4]) that this algorithm converges to a local minimum of the log-likelihood (20).

In the precise case of a Gaussian mixture model, the two steps of the algorithm become

- **E-step**, computation of $t_{ik} := \mathbf{E}(z_{ik}|y_i, \theta)$

$$t_{ik} = \frac{\pi_k \phi(y_i; \theta_k)}{\sum_{l=1}^{K} \pi_l \phi(y_i; \theta_l)} \tag{23}$$

- **M-step**, update of the parameters

$$\widehat{\pi}_k = \frac{1}{n} \sum_{i=1}^{n} t_{ik}, \tag{24}$$

$$\widehat{\mu}_k = \frac{\sum_{i=1}^{n} t_{ik} y_i}{\sum_{i=1}^{n} t_{ik}}, \tag{25}$$

$$\widehat{\Sigma}_k = \frac{\sum_{i=1}^{n} t_{ik} (y_i - \mu_k)(y_i - \mu_k)^T}{\sum_{i=1}^{n} t_{ik}}. \tag{26}$$

Observe that if we impose the t_{ik} to be 1 when the patch i belongs to the group k and 0 otherwise, the M-step consists in inferring the parameters of the Gaussian models for the groups, while the E-step uses the knowledge of the inferred model to update the groups themselves. This model provides a better clustering of the patches than a K-means clustering with the Euclidean norm (which only produces isotropic clusters) and consequently should yield better denoising results. This idea is used in [20, 22, 11] and the GMM model on patches is also used in [23]. A straightforward implementation of the denoising with a GMM model on the patches gives the result in the first line of Figure 11. However, this inference of a GMM also strongly suffers from the curse of the dimensionality and algorithms such S-PLE [20] or HDMI [11, 10] propose to use Gaussian Mixture models with intrinsic lower dimensions in order to reduce the number of parameters to estimate, as detailed in the following section.

5.4 Inference in high dimension

The dimensions of the patch spaces are usually high, from $p = 9$ (for 3×3 patches) to $p = 100$ for 10×10 patches, or even higher. Estimating the parameters of Gaussian models (or GMM) in such high dimensional spaces is complex. When p is large, patches seen as points in $\mathbb{R}^p$ are essentially isolated, the euclidean distance and the notion of nearest neighbor become much less reliable than in low dimensional spaces [9]. These phenomena, known as the curse of dimensionality, cause difficulties to decide which patches should

be grouped together in a common Gaussian model. Besides, parametric models such as Gaussian Mixture Models in high-dimension are usually over-parametrized: the covariance matrix of a Gaussian model in dimension $p = 100$ contains 5050 different coefficients. They necessitate huge quantities of data to be estimated correctly. Indeed, the convergence of the sample covariance matrices to the true covariance matrix depends on the ratio between the number n of samples and the dimension p. More precisely, if n and p both tend toward infinity while $\frac{n}{p}$ tends toward a constant $c > 0$, the eigenvalues of the sample covariance matrix $\widehat{\Sigma}(n)$ do not necessarily converge towards the eigenvalues of the model covariance matrix (Marčenko-Pastur Theorem [14] describes the limit law of the empirical distribution of these eigenvalues).

A consequence of the curse of dimensionality is that clustering methods such as K-means of GMM are often disappointing in high dimension, or do not converge at all if p is too large. Solutions to circumvent these problems usually rely on dimension reduction, or regularization of the model parameters. For instance, if the sample covariance matrix Σ is singular of ill-conditioned, or is not definite positive, it is usual to add a small εI_p to it. This is the strategy followed by [12, 23]. In the case of Gaussian Mixture Models, another approach consists in assuming that the intrinsic dimension of the Gaussian is lower than p. This is the idea adopted in [20], where the groups intrinsic dimensions are heuristically fixed to 1 (flat regions), $\frac{p}{2}$ or $p-1$. A more involved method consists in inferring for each group its own intrinsic dimension [11] (see Figure 11). The corresponding parsimonious model assumes that each Gaussian of the mixture lives in its own specific subspace.

6 Conclusion

In this chapter, we have focused on patch priors for image denoising. As we have seen, assuming Gaussian and GMM priors on image patches is now quite common in the restoration literature. We have tried to provide a unified point of view for all of these methods, in order to underline their similarities and differences. Table 1 summarizes the main features of the methods mentioned in this chapter. We have also described some of their limitations, such as the inference difficulties in high dimension or the absence of invariance properties to geometric transformations. We did not discuss the computational cost of these approaches, but this point is clearly a critical issue for industrial applications.

References

[1] Cecilia Aguerrebere, Andrés Almansa, Julie Delon, Yann Gousseau, and Pablo Musé. A bayesian hyperprior approach for joint image denoising and interpolation, with an application to hdr imaging. *IEEE Transactions on Computational Imaging*, 2017.

[2] Cecilia Aguerrebere, Julie Delon, Yann Gousseau, and Pablo Musé. Study of the digital camera acquisition process and statistical modeling of the sensor raw data. *Preprint Hal 00733538*, 2012.

[3] S Awate and R Whitaker. Image denoising with unsupervised information-theoretic adaptive filtering. In *International Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, pages 44–51, 2004.

[4] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[5] 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.

Method	Grouping	Modeling	Dimension reduction	Remarks	Denoising	Aggregation
Global [8]	all patches	Gaussian models	no	-	Wiener/HT	Uniform
Local [8]	local grouping in the image space	Gaussian models	no	-	Wiener/HT	Uniform
K-means	k-means in the patch space	Gaussian models	no	-	Wiener/HT	Uniform
NL-bayes [12]	nearest neighbours in the patch space	Gaussian models	no	flat areas are treated separately	Wiener	Uniform
PLE [23]	GMM		no	MAP-EM algorithm	Wiener at each step of the MAP-EM algorithm	Uniform
S-PLE [20]	GMM		yes	fixed intrinsic dimensions	MMLE	Uniform
HDMI [11]	GMM		yes	estimation of the intrinsic dimensions	MMLE	Uniform
EPLL [24]	-	GMM	no	GMM parameters infered on an external base	Variational formulation	

Table 1: This table summarizes the main features of the different methods mentioned in this chapter. Each line refers to a patch-based denoising method and the reference paper where it has been introduced. The columns correspond to the different steps we discussed in this chapter.

[6] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on image processing*, 16(8):2080–2095, 2007.

[7] Charles-Alban Deledalle, Loïc Denis, and Florence Tupin. How to compare noisy patches? patch similarity beyond gaussian noise. *International journal of computer vision*, 99(1):86–102, 2012.

[8] Charles-Alban Deledalle, Joseph Salmon, Arnak S Dalalyan, et al. Image denoising with patch based pca: local versus global. In *BMVC*, volume 81, pages 425–455, 2011.

[9] Christophe Giraud. *Introduction to high-dimensional statistics*, volume 138. CRC Press, 2014.

[10] A. Houdard, C. Bouveyron, and J. Delon. Clustering en haute dimensions pour le débruitage d'image. In *XXVIième colloque GRETSI*, 2017.

[11] Antoine Houdard, Charles Bouveyron, and Julie Delon. High-Dimensional Mixture Models For Unsupervised Image Denoising (HDMI). preprint, August 2017.

[12] M. Lebrun, A. Buades, and J. M. Morel. A Nonlocal Bayesian Image Denoising Algorithm. *SIAM J. Imaging Sci.*, 6(3):1665–1688, September 2013.

[13] Markku Makitalo and Alessandro Foi. A closed-form approximation of the exact unbiased inverse of the Anscombe variance-stabilizing transformation. *IEEE transactions on image processing*, 20(9):2697–2698, 2011.

[14] Vladimir A Marčenko and Leonid Andreevich Pastur. Distribution of eigenvalues for some sets of random matrices. *Mathematics of the USSR-Sbornik*, 1(4):457, 1967.

[15] David Mumford and Agnès Desolneux. *Pattern theory: the stochastic analysis of real-world signals*. CRC Press, 2010.

[16] Erik Ordentlich, Gadiel Seroussi, Sergio Verdu, Marcelo Weinberger, and Tsachy Weissman. A discrete universal denoiser and its application to binary images. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, volume 1, pages I–117. IEEE, 2003.

[17] Nicola Pierazzo, Jean-Michel Morel, and Gabriele Facciolo. Multi-scale dct denoising. *Image Processing On Line*, 7:288–308, 2017.

[18] Joseph Salmon and Yann Strozecki. From patches to pixels in non-local methods: Weighted-average reprojection. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 1929–1932. IEEE, 2010.

[19] Afonso M Teodoro, Mariana SC Almeida, and Mário AT Figueiredo. Single-frame image denoising and inpainting using gaussian mixtures. In *ICPRAM (2)*, pages 283–288, 2015.

[20] Yi-Qing Wang and Jean-Michel Morel. SURE Guided Gaussian Mixture Image Denoising. *SIAM J. Imaging Sci.*, 6(2):999–1034, May 2013.

[21] Tsachy Weissman, Erik Ordentlich, Gadiel Seroussi, Sergio Verdú, and Marcelo J Weinberger. Universal discrete denoising: Known channel. *IEEE Transactions on Information Theory*, 51(1):5–28, 2005.

[22] Jianbo Yang, Xuejun Liao, Xin Yuan, Patrick Llull, David J Brady, Guillermo Sapiro, and Lawrence Carin. Compressive sensing by learning a gaussian mixture model from measurements. *IEEE Transactions on Image Processing*, 24(1):106–119, 2015.

[23] Guoshen Yu, Guillermo Sapiro, and Stéphane Mallat. Solving inverse problems with piecewise linear estimators: from gaussian mixture models to structured sparsity. *IEEE Trans. Image Process.*, 21(5):2481–99, May 2012.

[24] Daniel Zoran and Yair Weiss. From learning models of natural image patches to whole image restoration. In *2011 Int. Conf. Comput. Vis.*, pages 479–486. IEEE, November 2011.

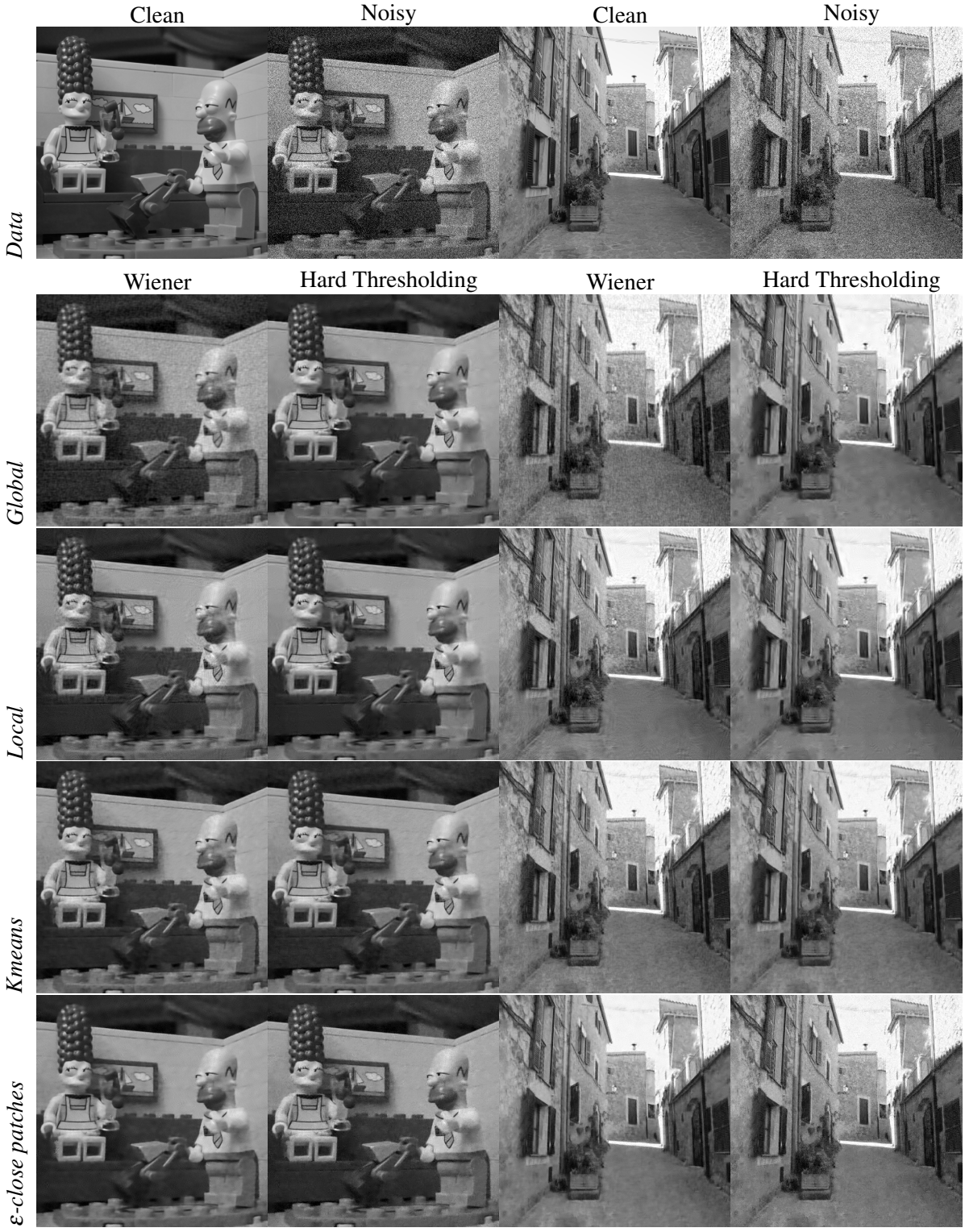

Figure 9: First line: two images and their noisy versions ($\sigma = 30$). Columns correspond to denoising strategies (Wiener or Hard thresholding). Lines correspond to grouping strategies: 1. one Gaussian model for all patches (PSNR, from left to right: 29.18dB, 31.22dB, 25.94dB, 26.85dB), 2. $K = 256$ local Gaussian models in the image space, see Figure 10 (PSNR, from left to right: 29.14dB, 30.72dB, 26.28dB, 26.88dB), 3. $K = 256$ local Gaussian models from a k-means clustering, see Figure 10 (PSNR: 31.30dB, 31.09dB, 26.92dB, 27.08dB), 4. local Gaussian models for group of ε-close patches (PSNR: 30.45dB, 29.65dB, 26.72dB, 25.95dB).

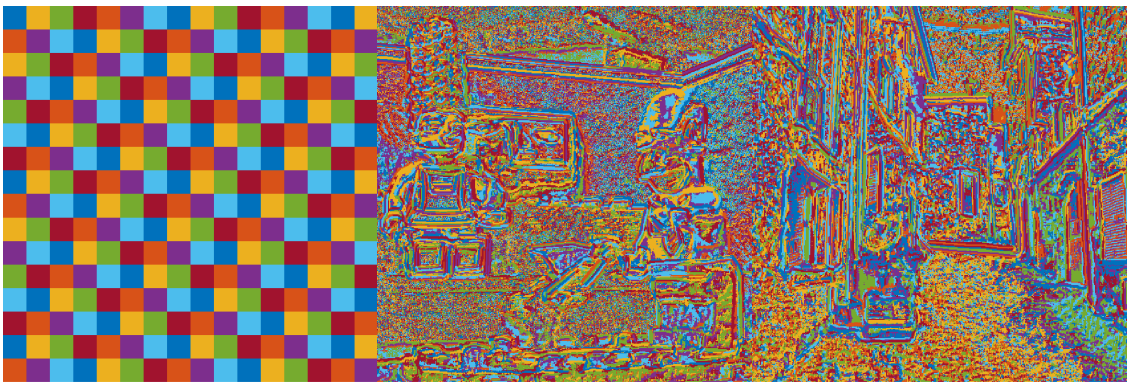

Figure 10: Left: the local grouping used in the local strategy. Middle and Right: the grouping used in the K-means strategy for the two images Simpson and Alley.

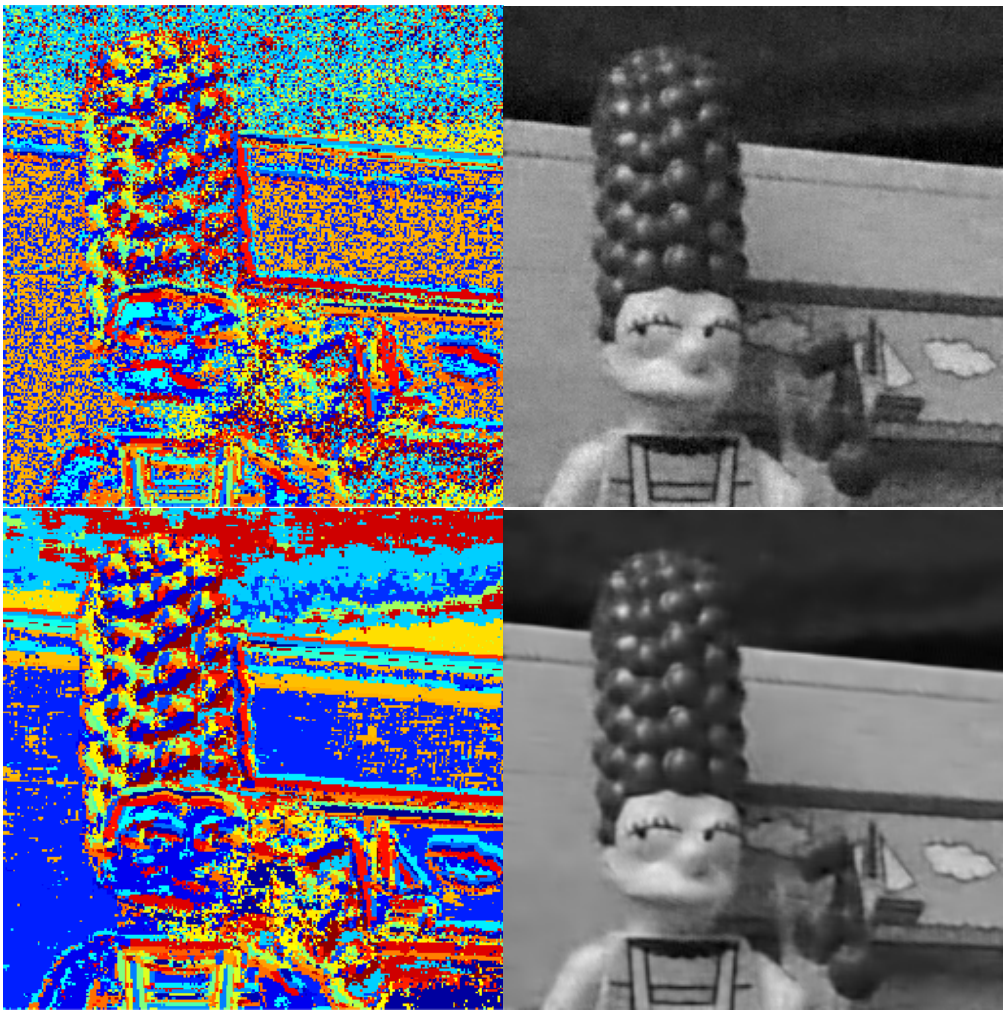

Figure 11: First line: Denoising with a full GMM model (50 groups) on all the patches. The clustering (left) is quite noisy and the denoising result (right) is not very good (PSNR: 28.50dB). Second line: Denoising with a GMM model (50 groups) with intrinsic dimension regularization as in [11]. The clustering (left) is smoother and the denoising yields quite good results (PSNR: 31.23dB)