A unified view on patch aggregation

Alexandre Saint-Dizier, Julie Delon, Charles Bouveyron

Noname manuscript No.
(will be inserted by the editor)

A unified view on patch aggregation

Alexandre SAINT-DIZIER · Julie DELON · Charles BOUVEYRON

the date of receipt and acceptance should be inserted later

Abstract Patch-based methods are widely used in various topics of image processing, such as image restoration or image editing and synthesis. Patches capture local image geometry and structure and are much easier to model than whole images: in practice, patches are small enough to be represented by simple multivariate priors. An important question arising in all patch-based methods is the one of patch aggregation. For instance, in image restoration, restored patches are usually not compatible, in the sense that two overlapping restored patches do not necessarily yield the same values to their common pixels. A standard way to overcome this difficulty is to see the values provided by different patches at a given pixel as independent estimators of a true unknown value and to aggregate these estimators. This aggregation step usually boils down to a simple average, with uniform weights or with weights depending on the trust we have on these different estimators. In this paper, we propose a probabilistic framework aiming at a better understanding of this crucial and often neglected step. The key idea is to see the aggregation of two patches as a fusion between their models rather than a fusion of estimators. The proposed fusion operation is pretty intuitive and generalizes previous aggregation methods. It also yields a novel interpretation of the Expected Patch Log Likelihood (EPLL) proposed in [29].

A. SAINT-DIZIER · J. DELON
MAP5, Université Paris Descartes, 12 Rue de l'École de Médecine, 75006 Paris, France

C. BOUVEYRON
Laboratoire J.A. Dieudonné, Université Côte d'Azur, Parc Valrose, 28 Avenue Valrose, 06108 Nice, France

Introduction

Image restoration and image editing remain two very active research topics, with numerous and various applications such as computational photography, video and multimedia processing, medical imaging or astronomical imaging, to mention just a few. Over the past fifteen years, a huge proportion of image restoration and editing approaches have been relying on patch-based representations. These representations exploit the semi-local redundancy of images and have led to decisive improvements in solving ill-posed inverse problems such as, for instance, denoising, inpainting or texture synthesis. Patches are small image pieces, and most of the time square pieces of size $\sqrt{d} \times \sqrt{d}$, with $\sqrt{d}$ between 3 and 20. They can be seen as vectors of $\mathbb{R}^d$. While the standard size of the images to be processed today is around 20 million pixels (or much more in medical or satellite imaging), working with patches is a way to work with data of reasonable dimension and variability, while capturing efficiently most of the image texture and geometry. In patch spaces, it also becomes possible to model data by simple mixtures of parametric distributions, which make Bayesian formulations much better posed and usable in practice. All of these patch-based methods require a projection (or aggregation) step to recreate an image from a set of processed patches. Indeed, overlapping patches do not necessarily share the same values on their common pixels after processing. Aggregation techniques aim at combining all these different overlapping patches into a single image. The goal of this paper is to propose a common and unique framework for this aggregation step which, while crucial, is not much discussed in the literature.

The first patch-based methods appear about twenty years ago in texture synthesis [8] and a few years later

in image restoration [3]. Since then, the use of patches in image editing has become very common, for texture synthesis [16], texture transfer [7], image enhancement [13] or more recently for the very fashionable problem of style transfer [11, 10]. In image restoration, patch-based methods are now used in almost all applications and it is reasonable to say that these methods have revolutionized the field, even if approaches based on neural networks [28] are now very serious rivals. In image denoising, most state of the art approaches rely on patches [2, 4, 29, 17, 18, 25, 14] and this is also the case in image and video inpainting [19, 25], image deblurring [5] or for many other inverse problems [27, 1].

All of these methods have in common to decompose images in overlapping patches and to make these patches collaborate for restoration, synthesis or editing purposes. Finally, the processed patches are merged together into a single image. While much attention has been paid on statistical or geometrical patch representations and interpretation, little work has been dedicated to explore the aggregation step. Going from the image space to the patch space is a linear and straightforward operation, but recovering an image from a set of overlapping patches is straightforward only if all of these patches share the same values on their common pixels. Even for patches coming from the same image, this property is lost as soon as the patches undergo non trivial operations. Each pixel belongs to d different patches and these patches yield d different estimates for the pixel value, as illustrated by Figure 1. In the literature, there are essentially four ways to answer the aggregation question:

1. For each pixel, keep only the estimator provided by the patch centered at this pixel;
2. For each pixel, average the d estimators with uniform weights (UWA);
3. For each pixel, average the d estimators with adapted weights (AWA);
4. Reconstruct the image from the patches as a solution of a variational problem.

The first solution is the one chosen in the first version of Non Local Means [3]. This approach ignores the information available in the rest of the patches. As a result, when applied in the context of image denoising for instance, residual noise can often be observed around edges or rare regions. A huge majority of methods tackle this issue by averaging the d estimates of the pixel, either with uniform weights [15] or with weights taking into account the precision of each estimator [4, 23], in order to minimize the variance of the aggregated estimator. For instance, the famous BM3D algorithm [4] uses weights which are chosen inversely proportional to the total variance of the sample of noisy patches used

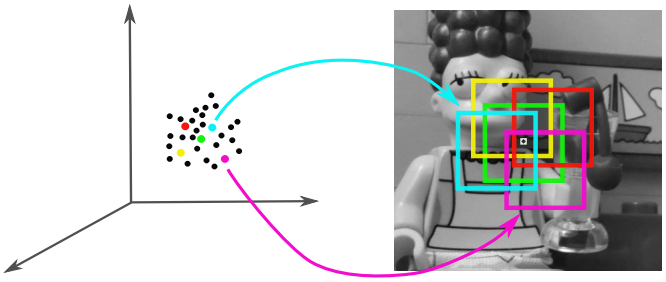


Fig. 1: Illustration of the aggregation step. Left: the patch space. Right: the reprojection onto the image space

to estimate the denoised patch. More recently, the multiscale DCT denoising described in [20] uses weights chosen inversely proportional to the number of non-zero coefficients of the DCT after thresholding, giving more weights to patches that have a lot of coefficients set to 0 (flat patches for example). The same idea is theorized and studied in [12]. Instead of the variance, some papers also attempt to minimize the risk of the final estimator at each pixel, by making use of Stein's Unbiased Risk Estimator (SURE) [6, 26]. The last solution is for instance explored in [9, 29] and consists in a global variational formulation of the restoration problem, including a global prior. These global formulations intrinsically include the aggregation problem, which is treated iteratively during the optimization process. In [29], the log of the global prior (the expected patch loglikelihood, or EPLL) is a sum of local priors on the patches and interpreted, up to a scalar, as "the expected log likelihood of a randomly chosen patch in the image". However, it can also be interpreted, up to a constant, as (the log of) a global image law, as already noted by [24]. Other attempts have been made to construct a global image law from local patch priors, such as the field of experts [22] which uses Markov Random Fields priors on pixels. We will see that the approach developed in the current paper has strong links with these global interpretations. In texture synthesis, alternatives to aggregation have been considered, such as [7] which finds a minimal error boundary cut between two overlapping patches, or [21] which uses conditioning to force the new patches to be coherent with the part of the image which has already been synthetised.

In this paper, we propose a novel perspective on the aggregation stage. To this aim, we focus on the case where each image patch is given a stochastic model on $\mathbb{R}^d$, for instance a Gaussian law or a Mixture of Gaussians. This situation is quite classical in Bayesian image restoration, where each patch is restored with a prior model [29, 17, 27, 25, 14]. It is usual that these different

models do not coincide on overlapping patches. In order to make these models coincide, we introduce the notion of *patch fusion*, which can be used to construct a global model on the whole image, up to a normalization. As we shall see, the classical aggregation techniques described above can be interpreted as special cases of this fusion framework. Our notion of patch fusion also reconciles the point of view developed in EPLL [29] and the conditioning approach suggested in [21] for texture synthesis.

The paper is organized as follows. In Section 1, we present the notions of patch model, patch agreement and patch fusion. We study the special cases of Uniform and Gaussian patch models in Section 2 and we show the links between our framework and the classical aggregation methods in Section 3. Section 4 gives some computational details. Finally, Section 5 is devoted to experiments and highlights the behaviour of the different aggregation models.

Notation

In this paper, random variables are denoted with uppercase letters, and their values are denoted with lowercase letters. X for instance, can take the value $x \in \mathbb{R}$. We denote by $p(x) = p(X = x)$ indifferently the probability of the event $\{X = x\}$ or the value of the density of X at x, whether X is discrete or continuous. When we are dealing with multidimensional spaces, d refers to the dimension of the whole space. Besides, for the sake of simplicity, d will be considered to be a perfect square, with $p \in \mathbb{N}$ such that $d = p^2$. Id refers to the identity matrix.

For two real numbers x and y, $[x, y]$ refers to the set of all real numbers between x and y, and $[\![x, y]\!]$ refers to the set of all integers between x and y.

$\mathcal{N}(\mu, \Sigma)$ refers to a Gaussian distribution, with mean μ and covariance Σ. If ν is a probability distribution, then $X \sim \nu$ means that X follows the law given by ν.

Considering an application $\phi : E \to F$ and $G \subset E$, we denote by $\phi|_G$ the restriction of ϕ to G. If x is a one (or two) dimensional vector in $\mathbb{R}^d$ (resp. $\mathbb{R}^{d^2}$) and G a subset of $[\![1, d]\!]$ or (resp. $[\![1, d]\!]^2$), we denote by $x|_G$ the restriction of x to the coordinates indexed by G (seeing x as an application from $[\![1, d]\!]$ or $[\![1, d]\!]^2$ to $\mathbb{R}$).

If $\phi : E \to F$ is an injective function, we denote by ϕ^{-1} the inverse of ϕ seen as an application from $Im(\phi)$ to E.

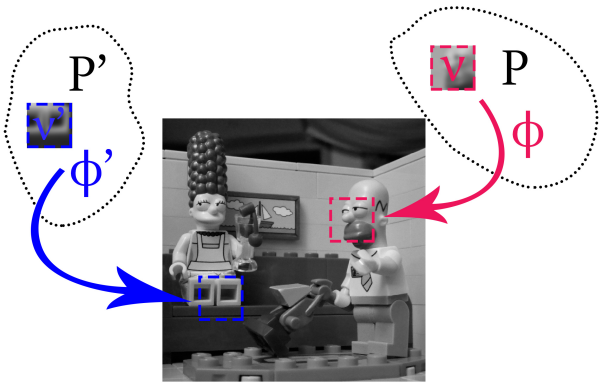


Fig. 2: Illustration of the proposed definition of a patch model

1 Patch model, agreement and fusion

In this section, we propose a framework to define properly a notion of patch fusion. This notion is motivated by the common situation encountered in many patch-based algorithms (dedicated to image restoration or image editing), where each patch is assumed to follow a prior distribution and the distributions of overlapping patches do not agree with each other. In the aggregation step, one can hope to be able to use all the information given by these priors. To address this issue, we define the notion of *patch model*. This enables to define a fusion between patch models, which, as we shall see, leads to a convenient and intuitive aggregation procedure, generalizing classical aggregation methods.

1.1 Patch model

We define in the following a patch model of size d on the discrete grid $\boldsymbol{\Omega}$ as a set of d positions on $\boldsymbol{\Omega}$ and a probability distribution ν on $\mathbb{R}^d$. For the sake of simplicity, we shall fix here once and for all the grid $\boldsymbol{\Omega}$, and we consider that the pixel of $\boldsymbol{\Omega}$ are ordered, i.e. we fix a bijective application $\boldsymbol{\phi} : \boldsymbol{\Omega} \to [\![1, |\boldsymbol{\Omega}|]\!]$.

Definition 1 A patch model P on the grid $\boldsymbol{\Omega}$ is a couple (Ω_P, ν_P), where $\Omega_P \subset \boldsymbol{\Omega}$ and ν_P is a probability distribution on $\mathbb{R}^{|\Omega_P|}$. We refer to ν_P as the *distribution* of the patch model, and to Ω_P as its *domain*.

We denote by $\mathcal{P}$ the set of all patch models on $\boldsymbol{\Omega}$ and by $\mathcal{P}_d$ the set of all patch models of size d. Clearly, $\mathcal{P} = \bigcup_{d \leq s_x \times s_y} \mathcal{P}_d$.

This definition can be seen as a generalization of the classical definition of a patch on a grid. Indeed, a traditional (deterministic) patch of size d on the grid $\boldsymbol{\Omega}$

is usually described by a vector P in $\mathbb{R}^d$ (which can be seen as the Dirac distribution δ_P in our definition) and a set of d pixels in $\boldsymbol{\Omega}$. Most of the time in the literature, this set of d pixels is assumed to be a square in $\boldsymbol{\Omega}$, or at least to form a connected set of pixels. We do not impose any constraint of the sort in our definition.

Proposition 1 *For $\Omega_P \subset \boldsymbol{\Omega}$, there exists a unique application $\phi_P : \Omega \to [\![1, |\Omega|]\!]$ which conserves the order of $\boldsymbol{\Omega}$, i.e. $\forall i, j \in \Omega_P, [\boldsymbol{\phi}(i) < \boldsymbol{\phi}(j) \Rightarrow \phi_P(i) < \phi_P(j)]$.*

Because we fix an order on $\boldsymbol{\Omega}$, there is no ambiguity in the correspondence between the domain Ω_P and the distribution ν_P of a patch model. A pixel $p \in \Omega_P$ corresponds to the $\phi_P(p)$-th coordinate in $\mathbb{R}^{|\Omega_p|}$. The goal of this framework is to be able to interpret ν_P as a distribution on the pixels of Ω_P just as well as a distribution on $\mathbb{R}^{|\Omega_p|}$.

Definition 2 (Restriction to a set of pixels) Let Ω_1, Ω_2 such that $\Omega_1 \subset \Omega_2 \subset \boldsymbol{\Omega}$, ϕ_1, ϕ_2 the corresponding applications (see proposition 1) and $x \in \mathbb{R}^{|\Omega_2|}$. We define $x|_{\Omega_1}^{\Omega_2}$, the restriction of x to Ω_1 with respect to Ω_2, by

$$x|_{\phi_2(\Omega_1)} \in \mathbb{R}^{|\Omega_1|},$$

where x is seen as an application from $[\![1, |\Omega_2|]\!] \to \mathbb{R}$.

Given ν a probability distribution on $\mathbb{R}^{|\Omega_2|}$, we define $\nu|_{\Omega_1}^{\Omega_2}$ as the marginal distribution of ν on $\phi_2(\Omega_1)$.

We are now in the position to define the notion of agreement between two patch models.

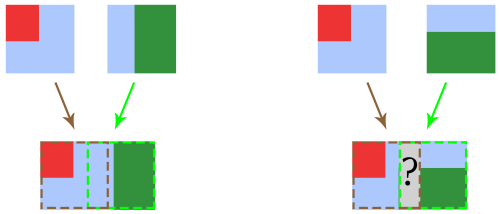
Definition 3 (Patch model agreement) Let $P_1 = (\Omega_1, \nu_1)$ and $P_2 = (\Omega_2, \nu_2)$ be two patch models in $\mathcal{P}$, and let $\Omega = \Omega_1 \cup \Omega_2$. We say that these two patch models agree and we write $P_1 \hat{=} P_2$ if and only if

$$\nu_1|_{\Omega_1 \cap \Omega_2}^{\Omega} = \nu_2|_{\Omega_1 \cap \Omega_2}^{\Omega}.$$

In other words, two patch models agree if they share the same distribution on their intersection. Therefore, two disjoint patch models (P and Q such that $\Omega_P \cap \Omega_Q = \emptyset$) agree automatically. The $\hat{=}$ relation is reflexive and symmetric, but not transitive.

Remark 1 This definition can be applied to deterministic patches. We say that they agree if their values on their overlap corresponds. We will also note this with the symbol $\hat{=}$.

We now define the notion of compatibility between patch models, which ensures that a set of patches, once restricted to the intersection of their domain, have distributions with compatible supports. This notion will be important in the definition of the patch model fusion in the next section.



(a) The two patches agree, the aggregation is straightforward

(b) The two patches do not agree, the aggregation is ambiguous

Fig. 3: Illustration of the notion of agreement between two deterministic patches

Definition 4 (Patch model compatibility) Let $(P_n)_{n \in [\![1,N]\!]} = (\Omega_n, \nu_n)_{n \in [\![1,N]\!]}$ be a set of N patch models. We say that these patch models are compatible if for any $K \leq N$, any subset $\{n_1, \dots, n_K\}$ of $[\![1, N]\!]$, and any subset V of $U = \cap_{k=1}^K \Omega_{n_k}$,

$$\cap_{k=1}^K \text{supp}(\nu_{n_k}|_V^{\Omega_{n_k}}) \neq \emptyset.$$

In the previous definition, $\text{supp}(\nu)$ denotes the support of the distribution ν, *i.e.* the complement of the largest open set on which this distribution vanishes. In practice, the notion of compatibility is not restrictive if we work with distributions with infinite support such as Gaussian distributions for instance.

1.2 Patch model fusion

We can now define the fusion of two patch models. As explained before, this definition is motivated by the usual situation encountered in patch-based algorithms where we end up with one or several distributions (and thus patch models) on the different patches, and we wish to deduce a distribution on the whole image. In the literature, such distributions are generally used to construct an estimator for each patch, such as the expectation or the maximum a posteriori, which leads to several different values for each pixel, and an aggregation procedure is used to deduce the final value of the pixels from these estimations. The patch model fusion permits to construct directly a distribution for the whole image from the different patch models, and thus to estimate the image without estimating the patches and losing the models information.

Definition 5 (Patch model fusion) Let $P_1 = (\Omega_1, \nu_1)$ and $P_2 = (\Omega_2, \nu_2)$ be two compatible patch models. We suppose that the distributions ν_1 and ν_2 have bounded densities f_1 and f_2.

The fusion $P_1 \odot P_2$ of P_1 and P_2 is the patch model defined by (Ω, ν) where $\Omega = \Omega_1 \cup \Omega_2$ and $\nu(dx) = f(x)dx$, with

$$\forall x \in \mathbb{R}^{|\Omega|}, \quad f(x) = \frac{f_1(x|^{\Omega}_{\Omega_1}) f_2(x|^{\Omega}_{\Omega_2})}{\int_{x \in \mathbb{R}^d} f_1(x|^{\Omega}_{\Omega_1}) f_2(x|^{\Omega}_{\Omega_2}) dx}.$$

Remark 2 – The compatibility between the patch models ensures that

$$0 < \int_{x \in \mathbb{R}^{|\Omega|}} f_1(x|^{\Omega}_{\Omega_1}) f_2(x|^{\Omega}_{\Omega_2}) dx.$$

– For the sake of simplicity, we restrict ourselves to the set of patch models with bounded densities. This strong assumption is convenient because it is stable for the fusion operation defined in the following, and it is always satisfied with the distributions we consider, but it could be relaxed. In practice, we only need to ensure that

$$\int_{x \in \mathbb{R}^{|\Omega|}} f_1(x|^{\Omega}_{\Omega_1}) f_2(x|^{\Omega}_{\Omega_2}) dx < +\infty.$$

The fusion operation simply consists in aggregating two patch models by merging their domains, and defining a distribution as a (particular) product of their original distributions. This definition has a very intuitive motivation, as we shall see in the next proposition.

Proposition 2 (Interpretation of the fusion) *Let $P_1 = (\Omega_1, \nu_1)$ and $P_2 = (\Omega_2, \nu_2)$ be two patch models and define $P_1 \odot P_2 = (\Omega, \nu)$. Assume that the distributions ν_1 and ν_2 have bounded densities f_1 and f_2. Without loss of generality, we suppose that pixels are ordered in such a way that the pixels of $\Omega_1 \setminus \Omega_2$ come first, then the pixels of $\Omega_1 \cap \Omega_2$ and eventually those of $\Omega_2 \setminus \Omega_1$.*

Let $Z \sim \nu_1$ and $T \sim \nu_2$ be two independent random variables. We write $Z = \begin{pmatrix} Z_{1\setminus 2} \\ Z_{1\cap 2} \end{pmatrix}$ where $Z_{1\setminus 2}$ are the $|\Omega_1 \setminus \Omega_2|$ first coordinates of Z and $Z_{1\cap 2}$ are its $|\Omega_1 \cap \Omega_2|$ other coordinates. We write also $T = \begin{pmatrix} T_{1\cap 2} \\ T_{2\setminus 1} \end{pmatrix}$ in the same way.

Then ν is the conditional probability distribution of the vector $\begin{pmatrix} Z_{1\setminus 2} \\ Z_{1\cap 2} \\ T_{2\setminus 1} \end{pmatrix}$ given $Z_{1\cap 2} = T_{1\cap 2}$.

Proof For $(z_{1\setminus 2}, z_{1\cap 2}, t_{2\setminus 1}) \in \mathbb{R}^{|\Omega_1 \setminus \Omega_2|} \times \mathbb{R}^{|\Omega_1 \cap \Omega_2|} \times \mathbb{R}^{|\Omega_2 \setminus \Omega_1|}$, we want to calculate

$$p\left(\begin{pmatrix} Z_{1\setminus 2} \\ Z_{1\cap 2} \\ T_{2\setminus 1} \end{pmatrix} = \begin{pmatrix} z_{1\setminus 2} \\ z_{1\cap 2} \\ t_{2\setminus 1} \end{pmatrix} | Z_{1\cap 2} = T_{1\cap 2} \right).$$

Now, this conditional distribution can be written

$$\frac{p(Z_{1\setminus 2} = z_{1\setminus 2}, Z_{1\cap 2} = z_{1\cap 2}, T_{2\setminus 1} = t_{2\setminus 1}, Z_{1\cap 2} = T_{1\cap 2})}{p(Z_{1\cap 2} = T_{1\cap 2})}$$

where

$$\begin{aligned} &p(Z_{1\setminus 2} = z_{1\setminus 2}, Z_{1\cap 2} = z_{1\cap 2}, T_{2\setminus 1} = t_{2\setminus 1}, T_{1\cap 2} = z_{1\cap 2}) \\ &= p(Z_{1\setminus 2} = z_{1\setminus 2}, Z_{1\cap 2} = z_{1\cap 2}) \times p(T_{2\setminus 1} = t_{2\setminus 1}, T_{1\cap 2} = z_{1\cap 2}) \\ &= f_1\left(\begin{pmatrix} z_{1\setminus 2} \\ z_{1\cap 2} \end{pmatrix} \right) \times f_2\left(\begin{pmatrix} z_{1\cap 2} \\ t_{2\setminus 1} \end{pmatrix} \right) = f_1(x|^{\Omega_1 \cup \Omega_2}_{\Omega_1}) \times f_2(x|^{\Omega_1 \cup \Omega_2}_{\Omega_2}) \end{aligned}$$

by independence of T and Z. Moreover,

$$\begin{aligned} p(Z_{1\cap 2} = T_{1\cap 2}) &= \int p(Z_{1\cap 2} = z_{1\cap 2}, T_{1\cap 2} = z_{1\cap 2}) dz_{1\cap 2} \\ &= \int p(Z_{1\cap 2} = z_{1\cap 2}) \times p(T_{1\cap 2} = z_{1\cap 2}) dz_{1\cap 2}. \end{aligned}$$

Since

$$\begin{aligned} p(Z_{1\cap 2} = z_{1\cap 2}) &= \int p(Z_{1\cap 2} = z_{1\cap 2}, Z_{1\setminus 2} = z_{1\setminus 2}) dz_{1\setminus 2} \\ &= \int f_1\left(\begin{pmatrix} z_{1\setminus 2} \\ z_{1\cap 2} \end{pmatrix} \right) dz_{1\setminus 2}, \end{aligned}$$

and

$$\begin{aligned} p(T_{1\cap 2} = z_{1\cap 2}) &= \int p(T_{1\cap 2} = z_{1\cap 2}, T_{2\setminus 1} = t_{2\setminus 1}) dt_{2\setminus 1} \\ &= \int f_2\left(\begin{pmatrix} z_{1\cap 2} \\ t_{2\setminus 1} \end{pmatrix} \right) dt_{2\setminus 1}, \end{aligned}$$

we conclude that

$$p(Z_{1\cap 2} = T_{1\cap 2}) = \int f_1(x|^{\Omega_1 \cup \Omega_2}_{\Omega_1}) \times f_2(x|^{\Omega_1 \cup \Omega_2}_{\Omega_2}) dx > 0.$$

□

The fusion operation is therefore a way to force two patch models to agree on their domain intersection while keeping most of the information contained in their models ν_1 and ν_2.

Proposition 3 *For any compatible patch models with bounded densities P, P' and P'' in $\mathcal{P}$, the fusion operation $\odot$ is well-defined and satisfies*

$P \odot P' \in \mathcal{P}$ and has a bounded density,

$$(P \odot P') \odot P'' = P \odot (P' \odot P''),$$

$$P \odot P' = P' \odot P.$$

Proof Let $P = (\Omega, f(x)dx)$, $P' = (\Omega', f'(x)dx)$, $P'' = (\Omega'', f''(x)dx)$ and $(\hat{\Omega}, \hat{f}(x)dx) = P \odot P'$. We have $\hat{\Omega} = \Omega \cup \Omega'$ and

$$\hat{f}(x) \propto f(x|_{\Omega}^{\hat{\Omega}}) \times f'(x|_{\Omega'}^{\hat{\Omega}}),$$

which clearly shows the commutativity. So $P \odot P'$ has also a bounded density and it is straightforward from the definition that $P \odot P'$ is compatible with P''. Besides, if we have $(\bar{\Omega}, \bar{f}dx) = (P \odot P') \odot P''$, we get

$$\bar{f}(x) \propto f(x|_{\Omega}^{\bar{\Omega}}) \times f'(x|_{\Omega'}^{\bar{\Omega}}) \times f''(x|_{\Omega''}^{\bar{\Omega}}),$$

which clearly shows the associativity. □

Remark 3 – This proposition ensures the stability and coherence of the operation, which can therefore be extended to any number of compatible patch models without ambiguity, and will be denoted $\bigodot_n P_n$, for any set $(P_n)_{n\in[\![1,N]\!]}$ of compatible patch models with bounded densities.

– For a set $(P_n)_{n\in[\![1,N]\!]} = (\Omega_n, f_n(x)dx)_{n\in[\![1,N]\!]}$ of compatible patch models with bounded densities, we have $\bigodot_n P_n = (\Omega, f(x)dx)$ with $\Omega = \bigcup_n \Omega_n$ and

$$\forall x \in \mathbb{R}^{|\Omega|}, f(x) \propto \prod_n f_n(x|_{\Omega_n}^{\Omega}).$$

We can merge the patch models in any order, we will always get the same result at the end (under the condition of compatibility and bounded density). This operation can be used to propagate and connect all the patch models to obtain a single image model.

1.3 Fusioned image model

The previous fusion operation can be used to define a global model on the whole image space from a set of all its local patch models.

Definition 6 Let E be a set of patch models. We say that E *covers the image support* if every pixel of $\boldsymbol{\Omega}$ belongs to the domain of at least one patch model of E, i.e.

$$\forall (i,j) \in \boldsymbol{\Omega}, \exists P = (\Omega, \nu) \in E \text{ such that } (i,j) \in \Omega.$$

We say that E *is coherent* if all patch models in E agree, i.e.

$$\forall\, (P_1, P_2) \in E^2, P_1 \hat{=} P_2.$$

We say that E *represents an image* if E covers the image support and is coherent.

This definition basically says that a set of patch models represents an image if all its patch models can be viewed as marginals of a large patch of size $|\boldsymbol{\Omega}|$.

For a set E of compatible patch models which covers the image support, Proposition 3 ensures that it is possible to fusion all the patch models of E to obtain a global model $(\boldsymbol{\Omega}, \nu) = \bigodot_{P\in E} P$ on the image. As a by-product, this constructs a new set $\hat{E}$ which represents the image, defined by

$$\hat{E} := \{\big(\Omega_P, \nu|_{\Omega_P}^{\boldsymbol{\Omega}}\big) \quad \text{with} \quad P \in E\}.$$

Remark 4 We can interpret this image model in the light of proposition 2. Consider E as a set of n compatible patch models, $I = (\boldsymbol{\Omega}, \nu_{\mathcal{I}})$ as a prior on the image, $\mathcal{Z}$ as a set of independent random variables following the probability distributions of the patch models of E and $\mathcal{I} \sim \nu_I$. Then, with a slight abuse of notation, we can write

$$\mathcal{I}|\,[\forall Z_1, Z_2 \in \mathcal{Z} \cup \{\mathcal{I}\}, Z_1 \hat{=} Z_2] \sim \nu_{\bigodot_{P\in E\cup\{I\}} P},$$

where $Z_1 \hat{=} Z_2$ means that Z_1 and Z_2 share the same values on the intersection of their domains.

2 Special cases

2.1 Uniform laws

A very simple example of patch model fusion can be derived in the case of uniform distributions.

Proposition 4 (Fusion of uniform patch models) *Let $A \subset \mathbb{R}^{|\Omega_A|}$ and $B \subset \mathbb{R}^{|\Omega_B|}$, and $P_A = (\Omega_A, \nu_A), P_B = (\Omega_B, \nu_B)$ be two patch models with uniform distribution on A and B,* i.e. *such that $\nu_A = \frac{1}{|A|}\mathbb{1}_A$, $\nu_B = \frac{1}{|B|}\mathbb{1}_B$. Let $\Omega = \Omega_A \cup \Omega_B$ and $C = \{x \in \mathbb{R}^{|\Omega|}; x|_{\Omega_A}^{\Omega} \in A \text{ and } x|_{\Omega_B}^{\Omega} \in B\}$.*

If $C \neq \emptyset$, then P_A and P_B are compatible and denoting $P_A \odot P_B$ by (Ω, ν), ν is a uniform distribution on C.

In other terms, the fusion of two uniform patch models is also a uniform patch model. Its distribution is the only uniform distribution on Ω whose marginals distributions on Ω_A and Ω_B are P_A and P_B.

This illustrates the behaviour of the fusion operation, which forces patch models to agree on their intersection. As a consequence, a patch model with a narrow distribution will impose its opinion to the other patch models, which is pretty intuitive: we except a confident model to be given more credit in the final aggregation. However, in the uniform case, this compromise is made quite brutally, since no patch model is ready to

accept an opinion out of its "reality", and this can easily result in compatibility issues when merging numerous patches. As we shall see, the Gaussian case keeps this behaviour, but in a softer way.

2.2 Gaussian laws

In several patch-based restoration algorithms such as NL-Bayes ([17]) or EPLL ([29]), patches are modelled by Gaussian distributions or Gaussian mixtures. The Gaussian distribution fits well with the previous framework, and also yields a closed form expression for the fusion operation.

Proposition 5 (Fusion of Gaussian patch models) *Let (Ω, ν) and (Ω', ν') be two Gaussian patch models with positive definite covariances:*

$$\nu = \mathcal{N}\left(\begin{pmatrix}\mu_x\\ \mu_y\end{pmatrix}, \begin{pmatrix}\Sigma_x & \Sigma_{xy}\\ \Sigma_{xy}^T & \Sigma_y\end{pmatrix}\right)$$
$$\text{and } \nu' = \mathcal{N}\left(\begin{pmatrix}\mu'_x\\ \mu'_y\end{pmatrix}, \begin{pmatrix}\Sigma'_x & \Sigma'_{xy}\\ \Sigma'^T_{xy} & \Sigma'_y\end{pmatrix}\right)$$

We suppose that the order on $\boldsymbol{\Omega}$ is such that the coordinates of their intersection, represented by x, come first.

Then (Ω, ν) and (Ω', ν') are compatible and the distribution of $(\Omega, \nu) \odot (\Omega', \nu')$ is Gaussian with parameters

$$\boldsymbol{\mu} = \begin{pmatrix}\mu_x\\ \mu_y\\ \mu_{y'}\end{pmatrix} + \begin{pmatrix}\Sigma_x\left(\Sigma_x + \Sigma'_x\right)^{-1}\\ \Sigma_{xy}^T\left(\Sigma_x + \Sigma'_x\right)^{-1}\\ -\Sigma'^T_{xy}\left(\Sigma_x + \Sigma'_x\right)^{-1}\end{pmatrix}(\mu'_x - \mu_x)$$

and,

$$\boldsymbol{\Sigma} = \begin{pmatrix}\begin{pmatrix}\Sigma_x & \Sigma_{xy}\\ \Sigma_{xy}^T & \Sigma_{y'}\end{pmatrix} & 0\\ 0 & \Sigma_{y'}\end{pmatrix} - \begin{pmatrix}\Sigma_x\left(\Sigma_x + \Sigma'_x\right)^{-1}\\ \Sigma_{xy}^T\left(\Sigma_x + \Sigma'_x\right)^{-1}\\ -\Sigma'^T_{xy}\left(\Sigma_x + \Sigma'_x\right)^{-1}\end{pmatrix}\begin{pmatrix}\Sigma_x\\ \Sigma_{xy}\\ -\Sigma'_{xy}\end{pmatrix}^T$$

Remark 5 In reality, we do not need the strict positiveness of the whole covariance, but we have to suppose that $\Sigma_x + \Sigma'_x$ is invertible, which is equivalent to the assumption that the two patch models are compatible.

Proof Let $Z = \begin{pmatrix}X\\ Y\end{pmatrix} \sim \nu$ and $Z' = \begin{pmatrix}X'\\ Y'\end{pmatrix} \sim \nu'$ be two independent Gaussian random vectors. From proposition 2, we know that we are looking for the conditional probability distribution of $\begin{pmatrix}X\\ Y\\ Y'\end{pmatrix}$ conditioned by $X = X'$.

Let $W = X - X'$. We know that W follows a Gaussian distribution, and $\mu_W = \mu_x - \mu'_x$ and $\Sigma_w = \Sigma_x + \Sigma'_x$. Similarly, we know that $\begin{pmatrix}Z\\ Y'\\ W\end{pmatrix}$ is a Gaussian random vector with parameters $\hat{\mu}, \hat{\Sigma}$ and we note $\hat{\mu} = \begin{pmatrix}\mu_z\\ \mu'_y\\ \mu_x - \mu_{x'}\end{pmatrix}$ and $\hat{\Sigma} = \begin{pmatrix}\Sigma_z & 0 & \Sigma_{ZW}\\ 0 & \Sigma'_y & \Sigma_{Y'W}\\ \Sigma_{ZW}^T & \Sigma_{Y'W}^T & \Sigma_W\end{pmatrix}$ (Z and Y' are independent). Let compute Σ_{ZW}:

$$\begin{aligned}\Sigma_{ZW} &= \mathbb{E}\left[ZW^T\right] - \mu_z[\mu_x - \mu_{x'}]^T\\ &= \mathbb{E}\left[ZX^T\right] - \mu_z\mu_x^T - \mathbb{E}\left[ZX'^T\right] + \mu_z\mu_{x'}^T\\ &= \begin{pmatrix}\Sigma_x\\ \Sigma_{xy}^T\end{pmatrix} + 0\end{aligned}$$

because of the independence of Z and X'. By symmetry we have $\Sigma_{Y'W} = \left(-\Sigma'^T_{xy}\right)$. So we can now use the classical result for the conditioning of Gaussian vectors:

$$\begin{aligned}\mu\begin{pmatrix}Z\\ Y'\end{pmatrix}_{|W=0} &= \mu\begin{pmatrix}Z\\ Y'\end{pmatrix} + \begin{pmatrix}\Sigma_{ZW}\\ \Sigma_{Y'W}\end{pmatrix}\Sigma_W^{-1}(0 - \mu_W)\\ &= \begin{pmatrix}\mu_Z\\ \mu_{Y'}\end{pmatrix} + \begin{pmatrix}\Sigma_x\left(\Sigma_x + \Sigma'_x\right)^{-1}\\ \Sigma_{xy}^T\left(\Sigma_x + \Sigma'_x\right)^{-1}\\ -\Sigma'^T_{xy}\left(\Sigma_x + \Sigma'_x\right)^{-1}\end{pmatrix}(\mu - \mu_X)\\ \Sigma\begin{pmatrix}Z\\ Y'\end{pmatrix}_{|W=0} &= \Sigma\begin{pmatrix}Z\\ Y'\end{pmatrix} - \begin{pmatrix}\Sigma_{ZW}\\ \Sigma_{Y'W}\end{pmatrix}\Sigma_W^{-1}\begin{pmatrix}\Sigma_{ZW}^T & \Sigma_{Y'W}^T\end{pmatrix}\\ &= \begin{pmatrix}\Sigma_Z & 0\\ 0 & \Sigma_{Y'}\end{pmatrix} - \begin{pmatrix}\Sigma_x\left(\Sigma_x + \Sigma'_x\right)^{-1}\\ \Sigma_{xy}^T\left(\Sigma_x + \Sigma'_x\right)^{-1}\\ -\Sigma'^T_{xy}\left(\Sigma_x + \Sigma'_x\right)^{-1}\end{pmatrix}\begin{pmatrix}\Sigma_x\\ \Sigma_{xy}\\ -\Sigma'_{xy}\end{pmatrix}^T\end{aligned}$$

□

As we can see, the set of all Gaussian patch models is stable by fusion. So if we have a set E of Gaussian patch models which covers the image, the resulting fusion of all the patch models from E will be a huge Gaussian model on the whole image support.

Proposition 6 *Let $(\Omega_n, \nu_n)_{n\in[\![1,N]\!]}$ be N Gaussian patch models, with means $(\mu_n)_{n\in[\![1,N]\!]}$ and precisions $(\Lambda_n)_{n\in[\![1,N]\!]}$. Let $P = (\Omega, \nu) = \bigodot_{n\in[\![1,N]\!]}(\Omega_n, \nu_n)$. For $n \in [\![1, N]\!]$, let ϕ_n be the application from Ω_n to $[\![1, |\Omega_n|]\!]$ associated to P_n, and ϕ the one associated to P (see proposition 1).*

Then P is a Gaussian patch model, and its precision Λ satisfies, for $i, j \in [\![1, |\Omega|]\!]$,

$$\Lambda(i,j) = \sum_{\substack{1\le n\le N\\ i,j\in\Omega_n}} \Lambda_n(\phi_n^{-1}\circ\phi(i), \phi_n^{-1}\circ\phi(j)). \quad (1)$$

Proof We know that $\nu(dx) = f(x)dx$ is Gaussian and we have, according to remark 3,

$$-\log f(x) \propto \sum_{n=1}^{N}(x|^{\Omega}_{\Omega_n} - \mu_n)^T\Lambda_n(x|^{\Omega}_{\Omega_n} - \mu_n).$$

Equation (1) follows by keeping for each pair (i,j) the indexes n such that pixels i and j belongs to Ω_n and by finding the indexes of these pixels in the patch Ω_n.

3 Link with classical aggregation methods

3.1 Standard aggregations

In the previous section, we have seen how to construct a distribution on a whole image from a set of compatible patch models. This construction, while theoretical, actually contains the main aggregation procedures used in the literature as special cases. More precisely, we shall see that these aggregation procedures can be seen as special cases of the fusion of Gaussian patch models with diagonal covariances.

Proposition 7 *Let (Ω,ν) and (Ω',ν') be two Gaussian patch models with diagonal positive definite covariances*

$$\nu = \mathcal{N}\left(\begin{pmatrix}\mu_x\\ \mu_y\end{pmatrix}, \begin{pmatrix}\Sigma_x & 0\\ 0 & \Sigma_y\end{pmatrix}\right)$$
$$\text{and } \nu' = \mathcal{N}\left(\begin{pmatrix}\mu'_x\\ \mu'_y\end{pmatrix}, \begin{pmatrix}\Sigma'_x & 0\\ 0 & \Sigma'_y\end{pmatrix}\right).$$

The variable x represents the common coordinates of the two patch models (μ_y and μ'_y may not have the same dimension). Then the patch models (Ω,ν) and (Ω',ν') are compatible and the distribution of $(\Omega,\nu)\odot(\Omega',\nu')$ is a Gaussian distribution with parameters

$$\boldsymbol{\mu} = \begin{pmatrix}(\Sigma_x^{-1}+\Sigma'^{-1}_x)^{-1}(\Sigma_x^{-1}\mu_x+\Sigma'^{-1}_x\mu'_x)\\ \mu_y\\ \mu_{y'}\end{pmatrix} \quad \text{and}$$

$$\boldsymbol{\Sigma} = \begin{pmatrix}(\Sigma_x^{-1}+\Sigma'^{-1}_x)^{-1} & 0 & 0\\ 0 & \Sigma_y & 0\\ 0 & 0 & \Sigma'_y\end{pmatrix}.$$

Moreover, the matrix $(\Sigma_x^{-1}+\Sigma'^{-1}_x)^{-1}$ is diagonal, and so is $\boldsymbol{\Sigma}$.

Proof This proposition is a direct application of proposition 5. □

The previous proposition states that if covariances of Gaussians are all supposed diagonal, then the resulting fusioned image has also a diagonal covariance. This boils down to assume that all the pixels are independent.

In the final image model, the mean at each pixel is simply a weighted average of all the means of the patches containing this pixel. The weights are given by the precisions of the marginals at these pixel. We recognize here the standard aggregation procedure usually called **variance aggregation**. The more precise an estimate is, the more it counts in the final estimate.

A more specific case is the one obtained when all covariances are identical and proportional to the identity matrix. In this case, the covariance of the resulting image model will be simply a diagonal, counting for each pixel the number of patches it belongs to. The resulting mean at a given pixel will be a simple average of all the means of the patches containing this pixel. This corresponds to the widely used **uniform aggregation**.

More formally, if we consider the limit case where each patch model has a covariance with infinite values except for its **central pixel**, we recover the most simple aggregation which consists in taking only the central pixel value as an estimate.

3.2 EPLL

More complex strategies including both patch restoration and aggregation into a single variational formulation have been considered in the literature. This is the case of the Expected Patch Log Likelihood (EPLL) of Zoran and Weiss [29]. Starting from an image

$$\tilde{I} = AI + \epsilon, \tag{2}$$

degraded by a linear operator A and an i.i.d. Gaussian noise $\epsilon \sim \mathcal{N}(0,\sigma^2 Id)$, the authors reconstruct a restored version as the solution of

$$\mathcal{D}(\tilde{I}) = \arg\min_I \frac{\lambda}{2}||AI-\tilde{I}||^2 - EPLL_f(I), \tag{3}$$

where $EPLL_f(I) = \sum_j \log\ f(x_j)$, with $\{x_j\}$ the set of all square patches of size $\sqrt{d}\times\sqrt{d}$ extracted from I and f a given prior density on the image patches.

The authors of [29] interpret the quantity $EPLL_f(I)$ as the empirical expectation of the log-likelihood of a patch (up to a multiplicative factor $\frac{1}{N}$ with N the number of patches). This quantity has another intuitive interpretation, as highlighted in the following proposition.

Proposition 8 *Let $\nu(dx) = f(x)dx$ be a given prior on patches of size $\sqrt{d}\times\sqrt{d}$. Let E be the set of all patch models of size $\sqrt{d}\times\sqrt{d}$ with distribution ν, i.e.*

$$E = \{(\Omega,\nu)|\exists(i,j)\in[\![1,s_x]\!]\times[\![1,s_y]\!],$$
$$\Omega = [\![i,i+\sqrt{d}]\!]\times[\![j,j+\sqrt{d}]\!]\}.$$

Let I be an image, seen as a vector of $\mathbb{R}^{|\boldsymbol{\Omega}|}$. Then if $\bar{P} = \bigodot_{P\in E} P = (\bar{\Omega},\bar{f}(x)dx)$ is well defined, we have

$$EPLL_f(I) = \log \bar{f}(I) + cst.$$

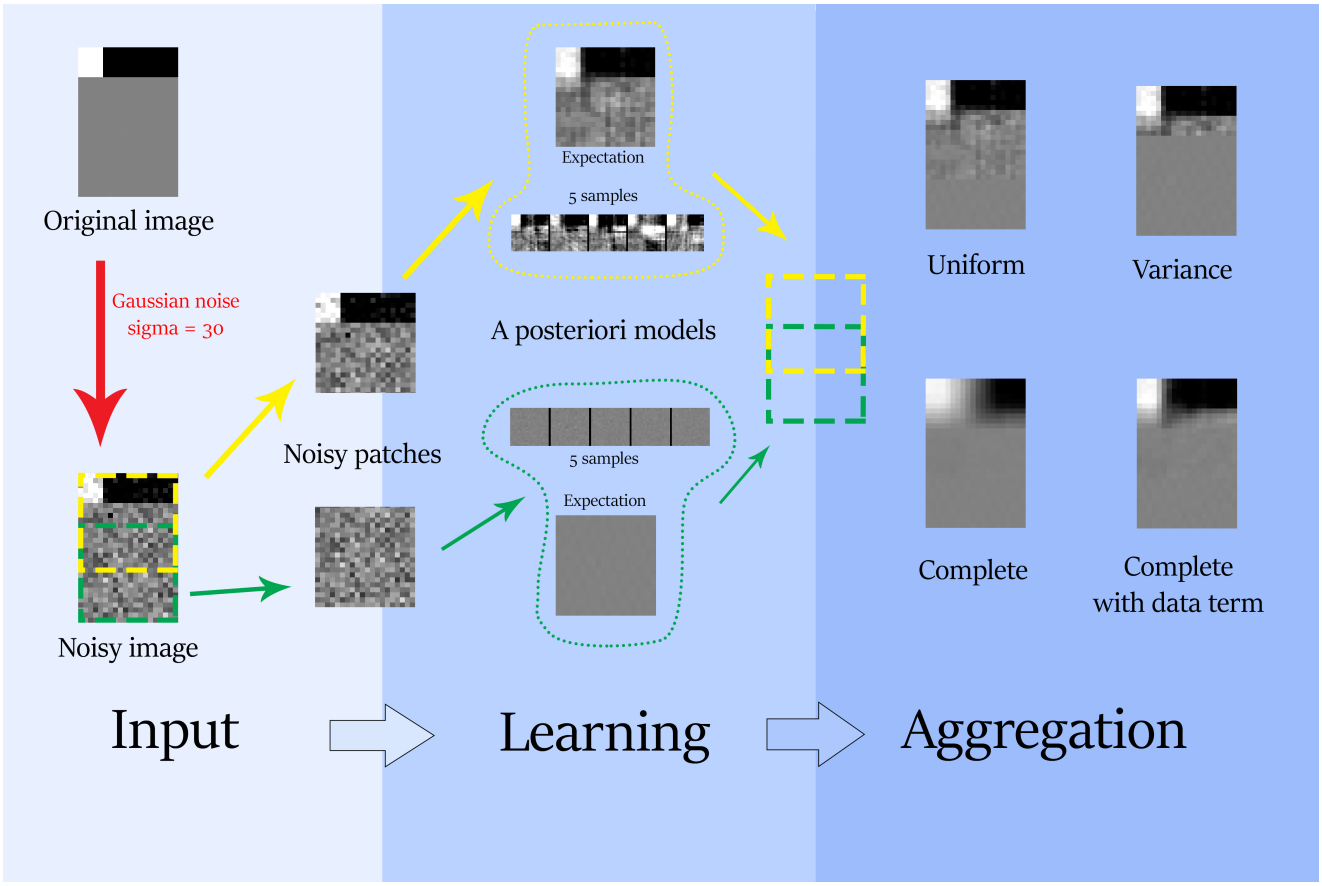


Fig. 4: Comparison of the different ways to aggregate two patch models.

The function $EPLL_f$ is the log of the density obtained by fusionning all square patch models on the grid with the same prior $f(x)dx$. Up to a constant, it can thus be interpreted as the *log* of a prior $p(I)$ on the whole image I. Consequently, by choosing $\lambda = \frac{1}{\sigma^2}$, the restoration operator $\mathcal{D}$ can be interpreted as a maximum a posteriori $\text{argmax}_I \log p(I|\tilde{I})$ on the whole image, since the term $-\frac{\lambda}{2}||AI - \tilde{I}||^2$ is, up to a constant, equal to $\log p(\tilde{I}|I)$ under the white Gaussian noise assumption.

Propositions 2 and 8 also clarify the link between the $EPLL$ approach and the iterative conditioning strategies used for instance in [21] for texture synthesis. Indeed, the fusioned image prior used in $EPLL$ can be interpreted as a law of a global random image obtained by aggregating all patch distributions and conditioning by their equality on all their intersections.

Now, consider the pure denoising case ($A = Id$). In this case, $\mathcal{D}$ can also be interpreted as a maximum likelihood for another fusioned distribution $\bar{\bar{f}}(x)dx$ on the whole image, as shown in the following distribution.

Proposition 9 *Keeping the notations of proposition 8, let $\bar{P} = \bigodot_{P \in E} P$ be the image model obtained by fusionning all patch models of E. Let $P_{\tilde{I}} = \left(\boldsymbol{\Omega}, \mathcal{N}(\tilde{I}, \frac{1}{\lambda} Id)\right)$ be an image model on the whole grid.*

Then if $(\boldsymbol{\Omega}, \bar{\bar{f}}(x)dx) := \bar{P} \odot P_{\tilde{I}}$, we have

$$\arg\min_I \frac{\lambda}{2}||I - \tilde{I}||^2 - EPLL_f(I) = \arg\max_I \bar{\bar{f}}(I).$$

Proof We just have to remark that

$$\begin{aligned}\log \bar{\bar{f}}(I) &= \log \bar{f}(I) + \log\left(e^{-\lambda \frac{\|I-\tilde{I}\|^2}{2}}\right) + cst \\ &= EPLL_f(I) - \frac{\lambda}{2}||I - \tilde{I}||^2 + cst. \square\end{aligned}$$

In the light of this proposition, the result of the EPLL algorithm in the denoising case is simply the maximum likelihood of the probability distribution obtained by merging all the patch models with a large Gaussian model centered on the noisy image and with variance $\frac{1}{\lambda}$.

Under the full degradation model (2), a last interpretation of the restoration operator $\mathcal{D}$ is possible, using the fusion of posterior patch models. To this aim, we have to assume that the degradation operator A is diagonal, which means that it acts separately on pixels.

The restriction of A to a domain Ω can thus be written $A_{|\Omega}$ and the model (2) restricted to Ω becomes

$$\tilde{I}_{|\Omega} = A_{|\Omega} I_{|\Omega} + \epsilon_{|\Omega}.$$

For a given patch model $P = (\Omega, f(x)dx)$ in E, the corresponding posterior patch model is just $(\Omega, f_{ap}(x)dx)$ where $f_{ap}(x)dx$ is the posterior obtained under this degradation model on Ω and the prior $f(x)dx$.

Proposition 10 *Keeping the notations of proposition 8, assume that each pixel of the $\mathbf{\Omega}$ is covered by exactly k patch models of E. For each patch model $P = (\Omega, f(x)dx)$ in E, we define the corresponding posterior patch model as $P_{ap} = (\Omega, f_{ap}(x)dx)$ with*

$$f_{ap}(x) \propto f(x) \frac{1}{(2\pi)^{d/2}\sigma^d} e^{-\frac{1}{2\sigma^2}\|A_{|\Omega}x - \tilde{I}_{|\Omega}\|^2}.$$

We define E_{ap} the set of all these posterior patch models,

$$E_{ap} = \{(\Omega, f_{ap}(x)dx), \textit{such that } (\Omega, f(x)dx) \in E\}.$$

Then, if $\bar{P}_{ap} = \bigodot_{P \in E_{ap}} P = (\boldsymbol{\Omega}, \bar{f}_{ap})$ is well defined, we have

$$\log \bar{f}_{ap}(I) = EPLL_f(I) - \frac{k}{2\sigma^2}||AI - \tilde{I}||^2 + cst.$$

Proof We just have to remark that

$$\sum_{P \in E} \|A_{|\Omega} I_{|\Omega} - \tilde{I}_{|\Omega}\|^2 = k||AI - \tilde{I}||^2. \qquad \square$$

In other words, for $\lambda = \frac{k}{\sigma^2}$, the operator $\mathcal{D}$ computes the maximum of a fusioned posterior model on the whole image, assuming that all patches have the same prior $f(x)dx$.

Remark 6 The hypothesis that each pixel is covered by exactly the same number of patch models can be ensured by supposing that the image is periodic. In practice, it is not satisfied when we consider all the overlapping deterministic patches. Yet, in this case, except for the pixels lying close to the image borders, the pixels are covered by exactly d patches.

4 Algorithms and computational details

Given a set of (compatible) patch models, we can use the fusion operation to obtain a global model on the union of their domains. In a general setup, this leads to generic algorithms which consist in fusionning all patch models of a set $\mathcal{P}$ iteratively, as proposed in algorithms 1 and 2. This procedure is stable and justified by proposition 3. How the fusion is performed in practice depends on the considered distributions. In the case of Gaussian or uniform patch models there is a closed-form solution, but approximated schemes could be used for more complicated distributions.

Algorithm 1 Patch model aggregation

Input: Set of patch models $\mathcal{P}$
Output: Aggregated image model $\boldsymbol{I}$
1: Take $P_0 \in \mathcal{P}$
2: $\boldsymbol{I} \leftarrow P_0$
3: **for** $P \in \mathcal{P} \setminus \{P_0\}$ **do**
4: $\quad \boldsymbol{I} \leftarrow \boldsymbol{I} \odot P$
5: **end for**

Algorithm 2 Aggregation of the patch models, equivalent approach

Input: Set of patch models $\mathcal{P}$
Output: Aggregated image model $\boldsymbol{I}$
1: **while** $|\mathcal{P}| > 1$ **do**
2: $\quad (P_1, P_2) \leftarrow \mathcal{P}$.get_two_first_elements()
3: $\quad P \leftarrow P_1 \odot P_2$
4: $\quad \mathcal{P}$.add(P) ▷ We place the fusion at the end of the queue
5: **end while**
6:
7: $P \leftarrow \mathcal{P}$.get_first() ▷ Only one element remains

In practice, for generic models, keeping in memory and computing the covariance of the whole image is not always tractable, since it requires to deal with a $(s_x \times s_y)^2$ matrix. Still, if necessary, we can approximate the global covariance matrix by noticing that pixels which are far enough from each other do not much influence each other. For instance, using standard Gaussian models for the image *Lena*, we observe that beyond a distance of $2\sqrt{d}$, patch models do not influence each other anymore (see figure 5). It means that the covariance matrix of the whole image is almost sparse. This enables to compute and store this covariance matrix much more easily, as described in algorithm 3.

In the case of Gaussian models, we know by proposition 6 that the precision matrix is sparse and this precision matrix can be computed and stored directly.

Observe that according to remark 3, the whole image distribution is, up to a constant, a generalization of the EPLL. If we only want to compute a MAP with this global model, we can use this equivalence to compute directly the maximum of this distribution with a simple optimization algorithm such as conjugate gradient or half quadratic splitting, as in [29].

5 Experiments

In this experimental section, we illustrate the behavior of the fusion for different models. For the sake of simplicity, we restrict our experiments to denoising problems. We also restrict our experiments to the case where

the patches of E are all square patches of size $\sqrt{d} \times \sqrt{d}$ in $\mathbf{\Omega}$.

5.1 Denoising with priors

In image denoising, in order to restore an unknown image I from its noisy version $I + \varepsilon$, we usually start by extracting all square patches $\{y_k,\ k \in \{1, \dots, |\mathbf{\Omega}|\}\}$ from $\tilde{I} = I + \varepsilon$. The noise model on patches can be written

$$y_k = x_k + \varepsilon_k,$$

with x_k the (unknown) patch before degradation. In the following, we will assume that the noise is i.i.d Gaussian of variance σ^2.

In this situation, bayesian patch-based methods use a common restoration framework to restore I from $I+\varepsilon$:

1. **estimation**: estimate a prior density f_k for each clean patch x_k
2. **restoration**: compute a denoised version $\hat{x}_k$ from y_k using the knowledge of the noise model and the prior f_k
3. **aggregation**: reconstruct a whole denoised image $\hat{I}$ from the set of denoised patches $\{\hat{x}_k,\ k \in \{1, \dots, |\mathbf{\Omega}|\}\}$.

The restoration step can for instance take the form of a maximum a posteriori

$$\hat{x_k} = \text{argmax}_x \frac{1}{2\sigma^2} \|y_k - x\|^2 - \log f_k(x).$$

Several methods in the literature use the previous restoration scheme, with slight variations. In the following, we will focus on three of them, which are representative of different choices in the three previously mentioned steps:

- NL-Bayes, [17], which estimates a specific Gaussian model $\mathcal{N}(\mu_k, \Sigma_k)$ for each patch x_k;
- HDMI, [14], which estimates a low-dimensional Gaussian Mixture model for the whole set of patches $x_k,\ k \in \{1, \dots, |\mathbf{\Omega}|\}$;
- EPLL, [29], which estimates a Gaussian Mixture model for patches on an external database, and replaces steps 2 and 3 above by the variational problem (3).

All of these methods yield a prior model f_k for each patch x_k. In the case of Gaussian Mixture Models, for the sake of simplicity, we choose to keep as a prior for x_k the Gaussian of the mixture which is the most likely for x_k.

Since the noise model is also Gaussian, these methods also yield Gaussian posterior models for each patch. We write these posteriors $\tilde{f}_k$, and

$$\tilde{f}_k(x|y_k) \propto f_k(x) e^{-\frac{\|x-y_k\|^2}{2\sigma^2}}.$$

In the following, we will illustrate how these priors or posterior models can be fusioned using the framework introduced in the previous sections. If we compute a fusioned prior model, the maximum a posteriori under the noise degradation model can be used to restore the image. In other words, if $\bar{f}$ is the fusioned image model density, the restored image is computed as the solution of

$$\text{argmin}_I \frac{1}{2\sigma^2} \|I - \tilde{I}\|^2 - \log \bar{f}(I). \tag{4}$$

If instead we compute a fusioned posterior model $\bar{f}(I|\tilde{I})$, the restored image can be computed directly as the maximum of this posterior, *i.e.*

$$\text{argmax}_I \bar{f}(I|\tilde{I}).$$

Now, writing x_k for the patches of I,

$$\begin{aligned} -\log \bar{f}(I|\tilde{I}) &= -\log \Pi_{k=1}^{|\mathbf{\Omega}|} f_k(x_k|y_k) \\ &= -\sum_{k=1}^{|\mathbf{\Omega}|} \log f_k(x_k) + \sum_{k=1}^{|\mathbf{\Omega}|} \frac{\|x_k - y_k\|^2}{2\sigma^2} \\ &= -\log \bar{f}(I) + d\frac{\|I - \tilde{I}\|^2}{2\sigma^2}. \end{aligned}$$

Observe that both strategies boil down to minimize an energy of the form

$$\text{argmin}_I \lambda \|I - \tilde{I}\| - \log \bar{f}(I), \tag{5}$$

with different values of λ. Taking the maximum of the fusioned posterior gives much more weight to the noisy image $\tilde{I}$ than computing the maximum a posteriori with the fusioned prior.

A schematic view of the different type of patch fusions (or aggregations) is proposed on figure 4.

Algorithm 3 Tractable approximation of the fusion procedure with sparsity hypotheses on the covariance matrix

Input: Set of square patches $\mathcal{P}$ of size $\sqrt{d}$, blocksize b
Output: Aggregated image $\boldsymbol{I}$

```
s ← 2 × √d
for B disjoint block of size b × b of the image do
    B̃ ← block of size (b + s) × (b + s) centered in B
    P_B̃ ← {P ∈ P | P ⊂ B̃}
    Compute I_B̃ using algorithm 1 or 2
    I|_B ← I_B̃|_B
end for
```

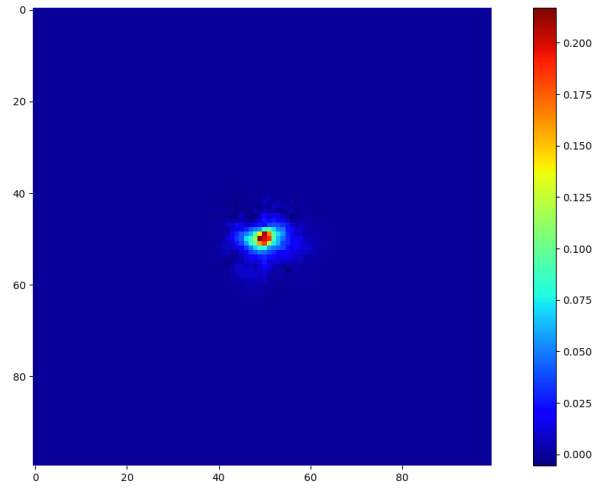


Fig. 5: In this experiment, we compute a complete Gaussian model for the image *Lena* thanks to our fusion algorithm. The figure shows the resulting correlation map of a pixel in a 100×100 patch. As we can see, the correlation decreases to 0 very fast when we move away from the center.

5.2 Results

Experiments are led on four different 512×512 images, *Lena*, *Barbara*, *Cartoon* and *Squares*. As we shall see, these images have different properties and the behaviour of the different fusion approaches strongly depend on their content.

Figures 6, 7 and 8 illustrate the use of different aggregation procedures on these four images and for the three mentioned algorithms (NL-Bayes in Figure 6, EPLL in Figure 7 and HDMI in Figure 8). The algorithms are used to compute local Gaussian priors for the different patches.

Different image models are then computed in order to restore the whole image. The first one corresponds to the standard uniform aggregation used in the literature. It starts by computing posteriors $\tilde{f}_k(x|y_k)$ on all patches. The means $\tilde{\mu}_k$ of these Gaussian posteriors are kept an averaged uniformly to yield the restored image. As we have seen, this gives the same result as replacing all covariance matrices by identity matrices, fusionning all these posterior models into a single image model and computing the maximum likelihood of this fusioned model. The results of this procedure are represented in the columns *Uniform aggregation* of the different figures.

The columns *Weighted aggregation* correspond to the aggregation with weights chosen inversely proportional to the variance of the patch model at each pixel. Again, it can be interpreted as a procedure where we fusion posterior models $\tilde{f}_k(x|y_k)$ on patches, approximating the covariance matrices by their diagonal. The resulting restored image is the mean of the fusioned posterior model.

Finally, the columns *Complete aggregation* in the different figures correspond to the complete fusion of the whole Gaussian posterior models, which is equivalent to maximizing the $EPLL$ with a weight λ equal to $\frac{k}{\sigma^2}$, as explained in proposition 10. The columns *Complete with data term* correspond to a higher lambda, whose value depends on the model used, in order to show how it influences the result.

NL-Bayes. The NL-Bayes algorithm infers many different Gaussian models and uses very small patches (5×5). As a consequence, most Gaussian covariances are quite well approximated by their diagonal, which explains why the different aggregation procedures only display minor differences. However, we can notice an improvement with the whole fusioned model on artificial images, in the corner of *Squares* for instance.

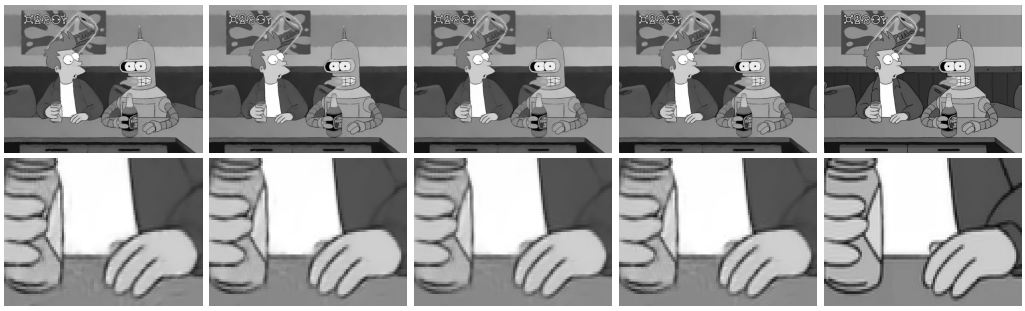
EPLL The EPLL model [29] makes use of 8×8 patches and learn a Gaussian mixture model with 200 groups on a large external set of images. One particularity is that the mean of patches are removed and the mean of Gaussians are imposed to be zero. In consequence, this set of models is not directly adaptable to our framework, since we need a model on each patch, and not on each of the averaged patches. In order to cope with this specificity, we removed the mean, select the most likely zero mean Gaussian in the GMM, and add to it the mean of the noisy patch. This approach might not be completely satisfying, but is still relevant to compare aggregation procedures. In practice, the weighted aggregation remains very similar to the uniform one (some tiny artifacts are removed), and the results of the full fusion are smoother, but with a strong "fluffy" effect. This effect is due to the fixed size of the patches and can be interpreted as a low frequency residual of the white Gaussian noise. Indeed, as this low frequency noise becomes part of the model, the complete aggregation increases it.

HDMI In HDMI [14] the model is learned on large patches (10×10) and with only a few dozen of low-dimensional Gaussians in the mixture. The different aggregations procedures produces more important differences. The uniform aggregation is efficient psnr-wise, but suffers from lots of artifacts. The whole fusioned model gives a better psnr for artificial images, and a

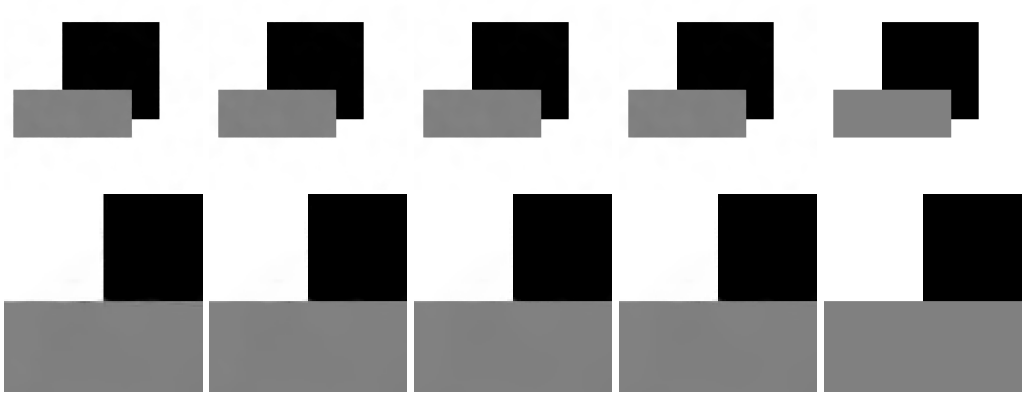
Uniform aggregation	Weighted aggregation	Complete aggregation	Complete data term $\lambda = 0.28$	Original image

(a) Psnr from left to right : 30.58, 30.49, 30.28, 30.66, ,∞

(b) Psnr from left to right : 28.99, 28.94, 28.83, 29.04,∞



(c) Psnr from left to right : 30.04, 29.98, 29.57, 30.35,∞



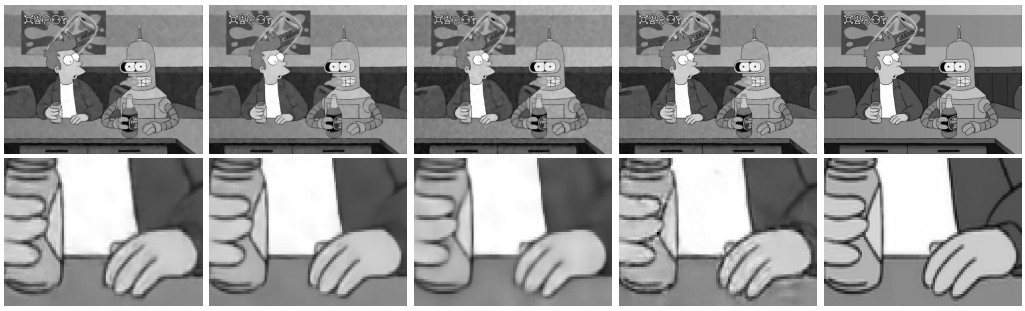
(d) Psnr from left to right : 45.28, 46.87, 47.35, 46.54,∞

Fig. 6: Comparison of the method on several images for NL-Bayes procedure for $\sigma = 30$.

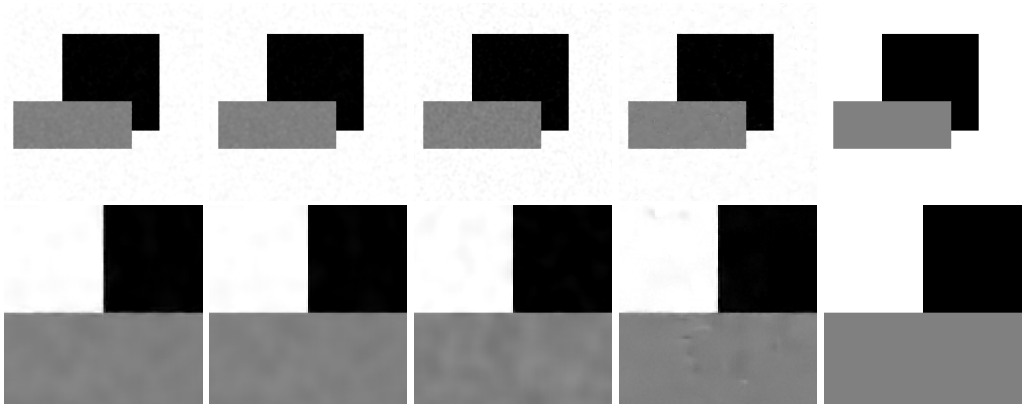
Uniform aggregation | Weighted aggregation | Complete aggregation | EPLL algorithm | Original image

(a) Psnr from left to right : 30.69, 30.42, 29.88, 30.71, ∞

(b) Psnr from left to right : 26.56, 26.18, 25.45, 27.55, ∞



(c) Psnr from left to right : 29.89, 29.62, 28.65, 30.49, ∞



(d) Psnr from left to right : 37.38, 39.09, 36.96, 39.51, ∞

Fig. 7: Comparison of the method on several images for the EPLL procedure for $\sigma = 30$.

large improvement in visual quality. Most of the artifacts are removed, and the image is much smoother. In particular, the "fluffy" effect present in EPLL in partially removed, as it does not belong to the model anymore. However, the results suffer from a lost of contrast (which may look like a lost of sharpness).

5.2.1 Precision map

Figure 9 shows the precision map obtained on *Lena* with the Gaussian models of the HDMI algorithm. As we can observe, the fusion yields good results in regions where the estimated model is confident (constant areas, regions with simple geometry). In other areas, it might be preferable to use a standard uniform aggregation instead. In practice, the results of these two procedures can be merged, using the precision map. Figure 9 shows the result of this merging, which clearly keep the best of both worlds.

5.2.2 Sparse aggregation

As shown in figure 4, the complete fusion aggregation needs fewer patches to obtain a visually satisfying result. The image can therefore be reconstructed using fewer patches, chosen either at random or using some heuristics to select the best models among them. This could be a way to speed up the learning phase, or to spend more time learning more complicated models. Figure 10 shows an example of a simple sparse aggregation, using only 4% of the patches (of size 10×10), so that each pixel belongs to only 4 patches.

5.3 Discussion

The visual differences between the different aggregation schemes increase with the size of the patches. In NL-Bayes [17], despite the variety of models, the use of 5×5 patches leads to very similar results for all the aggregation schemes. In contrast, important differences can be observed for these different schemes when used after EPLL [29] (8×8 patches) and HDMI [14] (10×10 patches).

The uniform aggregation is easy to compute and usually yields good PSNR results, but creates many artefacts in images since it averages uniformly all models, regardless of their estimated precision (contained in their covariance matrices). On the contrary, using the complete fusion of the different prior models an computing the MAP for the whole image removes most of the observed artefacts. This is quite understandable, since the method creates a model for which all the patches have to agree. Fusion results are very smooth and in return, they tend to suffer from a loss of contrast which makes the PSNR decrease. This is particularly obvious in regions where the model are not well-learned. Indeed, flat patch models tend to come with higher precisions than patch models representing geometric structures or contrasted textures. If, across an edge or a geometric structure, some patches are wrongly attributed to a flat patch model, this model will count significantly more than others in the fusion operation, which explains the observed contrast losses. These shortcomings can be reduced by increasing the weight λ of the data term in the final restoration (Equation 5), at the cost of a slight increase of noise. The variance aggregation usually offers a good compromise between the uniform aggregation and the complete fusion.

Another interesting point is the consequence of the fusion on the "fluffy effect" classically observed in single scale patch-based methods. When using HDMI [14], the fusion aggregation clearly reduces this defect, whereas it does increase it when using EPLL [29]. Indeed, in HDMI, patch priors have different means, which are already averages of different patches, whereas in EPLL, each noisy patch sees its own spatial average used as a mean for its model. Since these spatial averages contain all the low frequency noise of the original image, the whole fusioned model keeps intrinsically this low frequency noise and the results shown a very pronounced fluffy effect.

6 Conclusion and further work

We have presented here a new formal definition of a patch model, which permits to define the notion of agreement between overlapping patches. We have builded on this notion to propose a general common framework for the aggregation operation, seen as a fusion of different overlapping patch models. This common framework permits to gather and reinterpret all previous aggregation schemes used in the literature, and reduce the design of new ones to the design of a fusion operation.

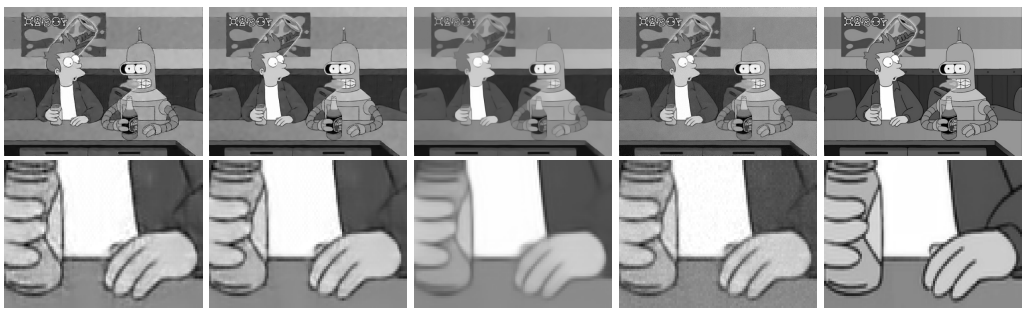
Our approach also permits to compute a fusioned image model which generalizes the Expected Patch Log Likelihood introduced by [29]. When patches are assumed to follow Gaussian distributions, this fusioned model is also Gaussian and its mean and covariance matrix can be computed by matrix multiplication. This whole fusioned model can in turn be used to restore the whole image. In practice, the fusion operation can be used for any model which leads to tractable computations.

We have compared experimentally several special cases of this fusion operation for patch-based image denoising. As we have seen, using the more complete

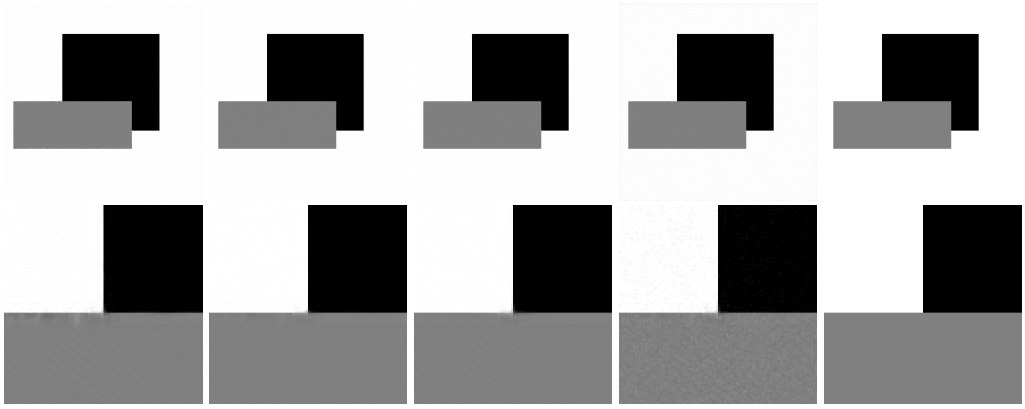
Uniform aggregation | Weighted aggregation | Complete aggregation | Complete with data term $\lambda = 1.1$ | Original image

(a) Psnr from left to right : 31.12, 31.10, 28.16, 29.96, ∞

(b) Psnr from left to right : 29.55, 29.54, 25.57, 28.72, ∞



(c) Psnr from left to right : 30.55, 30.52, 25.67, 29.34, ∞



(d) Psnr from left to right : 44.24, 48.77, 46.37, 35.62, ∞

Fig. 8: Comparison of the method on several images for the HDDC procedure for $\sigma = 30$.

(a) precision map (b) classic aggregation (c) conditionned aggregation (d) precision estimate, mix of (b) and (c)

Fig. 9: The inverse of the diagonal of the covariance matrix gives us the precision of the marginal of each pixel. This is a basic estimate of how sure the model is for each pixel. This inverse (prec) is represented on the left, followed by the classic aggregation (mean), the fusion aggregation (fusion) and an extended estimate : prec $\times$ fusion + (1-prec)$\times$ mean. The more confident we are in our estimate, the more we stay close to it.

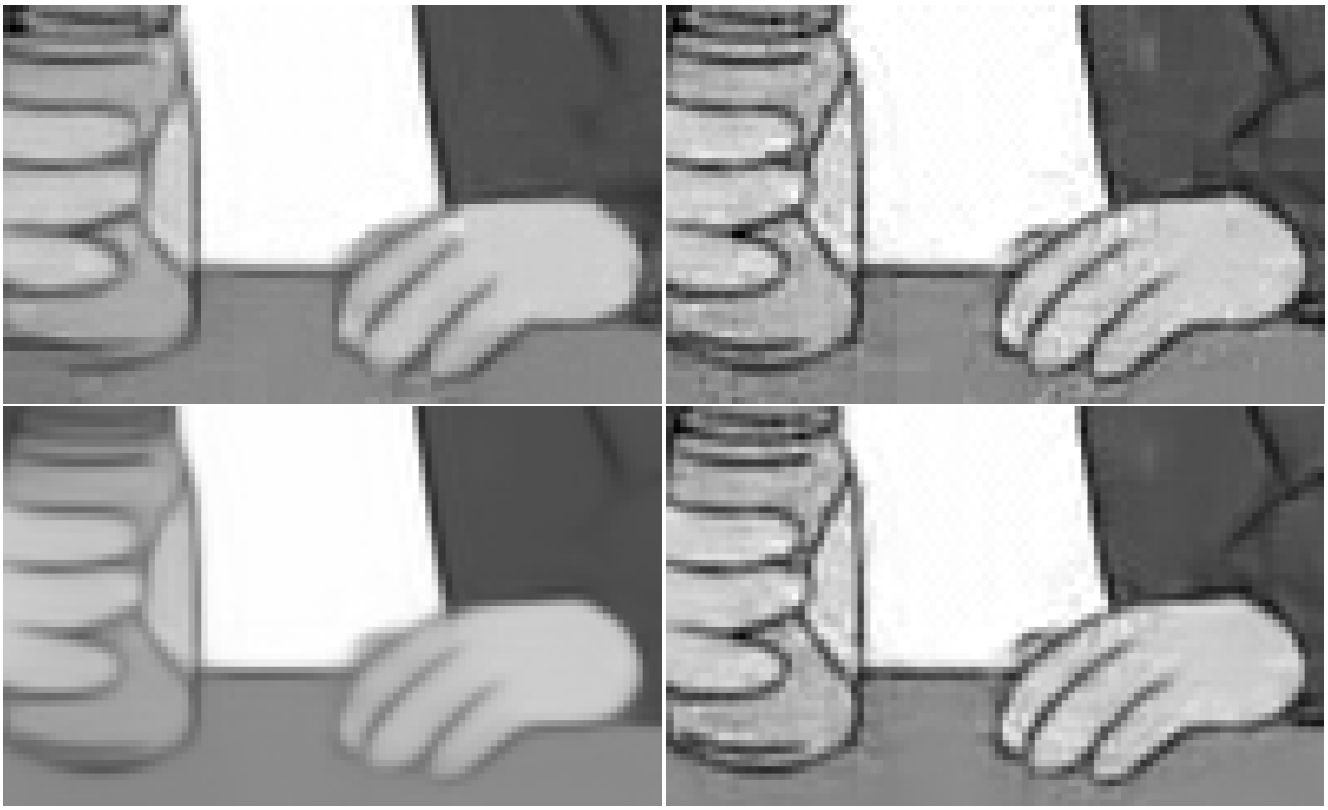


Fig. 10: Comparison of different aggregation procedures on the cartoon image. On the top row, only 4% of the patches are used, and on the bottom row, all the patches are used. On the left the complete aggregation, and on the right the uniform aggregation

fusion operation usually yields smaller PSNR results but highly reduces the visual artifacts. On very geometric and artificial images, it sometimes outperforms the standard uniform aggregation. The complete aggregation is preferable if the model is well-trained, since it takes advantage of all the provided information. However, patch models are practice far from perfect, in particular on natural images.

In the near future, we intend to apply this aggregation to wider type of inverse problems, and wider class of models, like Gaussian mixtures. We also plan to find a better fusion operation. Even if the proposed fusion is very intuitive, it has some limitations. For instance, when merging two Gaussian patch models, one could expect that the resulting covariance would depend on the means of the Gaussian models, but this is not the case with the proposed definition. It should be interesting to investigate more deeply in this direction, using ideas from the optimal transport theory.

Acknowledgment.

We would like to thank Arthur Leclaire and Pablo Arias for their insightful comments. This work has been partially funded by the French Research Agency (ANR) under grant nro ANR-14-CE27-001 (MIRIAM).

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. Michal Aharon, Michael Elad, and Alfred Bruckstein. *rmk*-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, 54(11):4311–4322, 2006.
3. Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. A review of image denoising algorithms, with a new one. *Multiscale Modeling & Simulation*, 4(2):490–530, 2005.
4. 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.
5. Aram Danielyan, Vladimir Katkovnik, and Karen Egiazarian. Bm3d frames and variational image deblurring. *IEEE Transactions on Image Processing*, 21(4):1715–1728, 2012.
6. Charles-Alban Deledalle, Vincent Duval, and Joseph Salmon. Non-local methods with shape-adaptive patches (nlm-sap). *Journal of Mathematical Imaging and Vision*, 43(2):103–120, 2012.
7. Alexei A Efros and William T Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 341–346. ACM, 2001.
8. Alexei A Efros and Thomas K Leung. Texture synthesis by non-parametric sampling. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1033–1038. IEEE, 1999.
9. Michael Elad and Michal Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image processing*, 15(12):3736–3745, 2006.
10. Michael Elad and Peyman Milanfar. Style transfer via texture synthesis. *IEEE Trans. Image Processing*, 26(5):2338–2351, 2017.
11. Oriel Frigo, Neus Sabater, Julie Delon, and Pierre Hellier. Split and match: Example-based adaptive patch sampling for unsupervised style transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 553–561, 2016.
12. Onur G Guleryuz. Weighted averaging for denoising with overcomplete dictionaries. *IEEE Transactions on Image Processing*, 16(12):3020–3034, 2007.
13. Yoav HaCohen, Eli Shechtman, Dan B Goldman, and Dani Lischinski. Non-rigid dense correspondence with applications for image enhancement. In *ACM transactions on graphics (TOG)*, volume 30, page 70. ACM, 2011.
14. Antoine Houdard, Charles Bouveyron, and Julie Delon. High-dimensional mixture models for unsupervised image denoising (hdmi). 2017.
15. Charles Kervrann and Jérôme Boulanger. Optimal spatial adaptation for patch-based image denoising. *IEEE Transactions on Image Processing*, 15(10):2866–2878, 2006.
16. Vivek Kwatra, Irfan Essa, Aaron Bobick, and Nipun Kwatra. Texture optimization for example-based synthesis. *ACM Transactions on Graphics (ToG)*, 24(3):795–802, 2005.
17. Marc Lebrun, Antoni Buades, and Jean-Michel Morel. A nonlocal bayesian image denoising algorithm. *SIAM Journal on Imaging Sciences*, 6(3):1665–1688, 2013.

18. Marc Lebrun, Miguel Colom, Antoni Buades, and Jean-Michel Morel. Secrets of image denoising cuisine. *Acta Numerica*, 21:475–576, 2012.
19. Alasdair Newson, Andrés Almansa, Matthieu Fradet, Yann Gousseau, and Patrick Pérez. Video inpainting of complex scenes. *SIAM Journal on Imaging Sciences*, 7(4):1993–2019, 2014.
20. Nicola Pierazzo, Jean-Michel Morel, and Gabriele Facciolo. Multi-scale dct denoising. *Image Processing On Line*, 7:288–308, 2017.
21. Lara Raad, Agnès Desolneux, and Jean-Michel Morel. A conditional multiscale locally gaussian texture synthesis algorithm. *Journal of Mathematical Imaging and Vision*, 56(2):260–279, 2016.
22. Stefan Roth and Michael J Black. Fields of experts. *International Journal of Computer Vision*, 82(2):205, 2009.
23. 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.
24. Sonia Tabti, Charles-Alban Deledalle, Loïc Denis, and Florence Tupin. Modeling the distribution of patches with shift-invariance: application to sar image restoration. In *Image Processing (ICIP), 2014 IEEE International Conference on*, pages 96–100. IEEE, 2014.
25. 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.
26. Dimitri Van De Ville and Michel Kocher. Sure-based non-local means. *IEEE Signal Process. Lett.*, 16(11):973–976, 2009.
27. Guoshen Yu, Guillermo Sapiro, and Stéphane 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.
28. Kai Zhang, Wangmeng Zuo, and Lei Zhang. Ffdnet: Toward a fast and flexible solution for cnn based image denoising. *IEEE Transactions on Image Processing*, 2018.
29. Daniel Zoran and Yair Weiss. From learning models of natural image patches to whole image restoration. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 479–486. IEEE, 2011.