HAL
open science

A unified view on patch aggregation

Alexandre Saint-Dizier, Julie Delon, Charles Bouveyron

To cite this version:

Alexandre Saint-Dizier, Julie Delon, Charles Bouveyron. A unified view on patch aggregation. 2018. hal-01865340v2

HAL Id: hal-01865340

https://hal.science/hal-01865340v2

Preprint submitted on 9 Jan 2019 (v2), last revised 6 Apr 2019 (v3)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

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 [40].

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

Over the past fifteen years, numerous image restoration and image 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 [3, 2, 6, 40, 24, 23, 39, 35, 19, 38], inpainting [25], interpolation [38, 35], deblurring [7], image enhancement [17], texture synthesis [11, 10, 22] style transfer [15, 13] or HDR imaging [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 this merging or 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 (*central aggregation*);
2. For each pixel, average the d estimators with uniform weights (*uniform aggregation*);

3. For each pixel, average the d estimators with adapted weights (*weighted aggregation*);
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 majority of methods tackle this issue by averaging the d estimates of the pixel, either with uniform weights [21] or with weights taking into account the precision of each estimator [6, 31, 34], in order to minimize the variance of the aggregated estimator. A recent approach [29], called SOS boosting, proposes to improve iteratively a denoising algorithm by reducing the gap between each restored patch and its value after uniform aggregation. The BM3D algorithm [6] uses weights which are chosen inversely proportional to the total variance of the sample of noisy patches used to estimate the denoised patch. More recently, the DCT-based denoising approaches [16, 27] use 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). Other approaches draw on similar ideas to derive optimal weights [9, 32, 20, 14]. Instead of the variance, some authors also attempt to minimize the risk of the final estimator at each pixel, by making use of Stein's Unbiased Risk Estimator (SURE) [8, 36]. In [4], a comparison is led between global optimization and weighted aggregation for denoising purposes.

The last solution for patch aggregation, explored for instance in [12, 40], 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 [40], the log of the global prior (the expected patch log likelihood, 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 probability distribution, as already noted by [33]. Other attempts [30, 5] have been made to construct a global image probability distribution from local patch priors, such as the field of experts [30] 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 a related direction, the fact that patches should coincide on their intersections can also be written as a hard constraint that can be included in any variational framework, as explored in the recent paper [26].

In texture synthesis, alternatives to aggregation have been considered, such as [10] which finds a minimal error boundary cut between two overlapping patches, or [28] which uses conditioning to force the new patches to be coherent with the part of the image which has already been synthesized.

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 distribution or a mixture of Gaussians. This situation is quite classical in Bayesian image restoration, where each patch is restored with a prior model [40, 23, 37, 38, 35, 19]. It is usual that these different models do not coincide on overlapping patches. In order to overcome this limitation, we introduce the notion of *patch fusion*, which draw on all the prior models to construct a global model on the whole image (up to a normalization), by taking into account the fact that these models should coincide on their overlaps. At the end, the final models for overlapping patches coincide but are not generally the same than the ones prescribed as input. As we shall see, the classical aggregation techniques described above can be interpreted as special cases of our fusion framework. Our notion of patch fusion also reconciles the point of view developed in EPLL [40] and the conditioning approach suggested in [28] for texture synthesis.

The paper is organized as follows. In Section 1, we recall the main steps of patch-based signal processing. In Section 2, 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 3 and we show the links between our framework and the classical aggregation methods in Section 4. Finally, Section 5 is devoted to experiments and highlights the behavior of the different aggregation models.

Notations

We define in the following some of the notations used in this paper.

- For a continuous random variable X, we denote by $p(X = x)$ the value of the probability density function of X at x.
- The identity map of the set S is denoted by Id_S, and by Id when the notation is not ambiguous.
- For two real numbers a and b, $[\![a, b]\!]$ refers to the set of all integers between a and b.
- Let E be a finite set and consider a map $\phi : E \to \mathbb{R}$, and F a subset of E. We denote by $\phi_{|F}$ the restric-

tion of the map ϕ to F. If ν is a probability distribution on $\mathbb{R}^E$, we define $\nu_{|F}$ the marginal distribution of ν on $\mathbb{R}^F$. If $F = \emptyset$, by convention, we define $\nu_{|F} = 1$.

1 Background: patch aggregation in image processing

The goal of this Section is to present the main steps of patch-based signal processing, as illustrated in Figure 1. We focus here on the case of images, which are 2D-signals, but all the concepts presented hereafter are much more general and can be defined for signals with any number of dimensions[1].

Let $\mathbf{\Omega}$ be a discrete rectangular grid of size $s_x \times s_y$ in $\mathbb{R}^2$ and let $u : \mathbf{\Omega} \to \mathbb{R}$ be a grey level image on $\mathbf{\Omega}$.

1.1 Patch extraction

A patch of an image u can be written as $x_P = u_{|\Omega_P} \in \mathbb{R}^{\Omega_P}$ where $\Omega_P \subset \mathbf{\Omega}$ is the domain of the patch. The number of pixels $|\Omega_P|$ is called the *size* of the patch. Patches usually considered in the literature have connected domains. For example, if $\Omega_P = [\![1, \sqrt{d}^2]\!]$, $u_{|\Omega_P}$ is the square patch of size d at the top left corner of u. The image u can be considered itself as a (large) patch of size $s_x \times s_y$, and contains $N = (s_x - \sqrt{d} + 1) \times (s_y - \sqrt{d} + 1)$ overlapping square patches of size d.

The *patch extraction* is characterized by an extraction operator E which gives a set of patches from an image. In most applications, this set is composed of all overlapping square patches of size d of u. Assimilating u to a $s_x \times s_y$ matrix and patches as vectors of size d (read column-wise), this extraction operator can for instance be written

$$E : \mathbb{R}^{s_x \times s_y} \to \mathbb{R}^{d \times N},$$

where the ith column y_i of the matrix $E(u)$ is the ith patch of u. Since E is a linear operator, $\mathrm{Im}(E)$ is a linear subspace of $\mathbb{R}^{d \times N}$ of dimension $s_x \times s_y$ at most. Therefore, $\mathrm{Im}(E) \neq \mathbb{R}^{d \times N}$.

In the general case, E is not surjective, and an element of $\mathrm{Im}(E)$ has lots of redundancies, since each pixel may appear in many different patches.

1.2 Patch-based editing or restoration

Given an extraction operator E returning a set of N patches, patch-based signal processing consists in processing the set of patches $E(u)$ instead of the signal u.

[1] For example, patches of size l of a 1D-signal $x = (x_1, \ldots, x_n)$ are of the form $(x_i, \ldots, x_{i+l})$, for $i \in [\![1, n-l]\!]$.

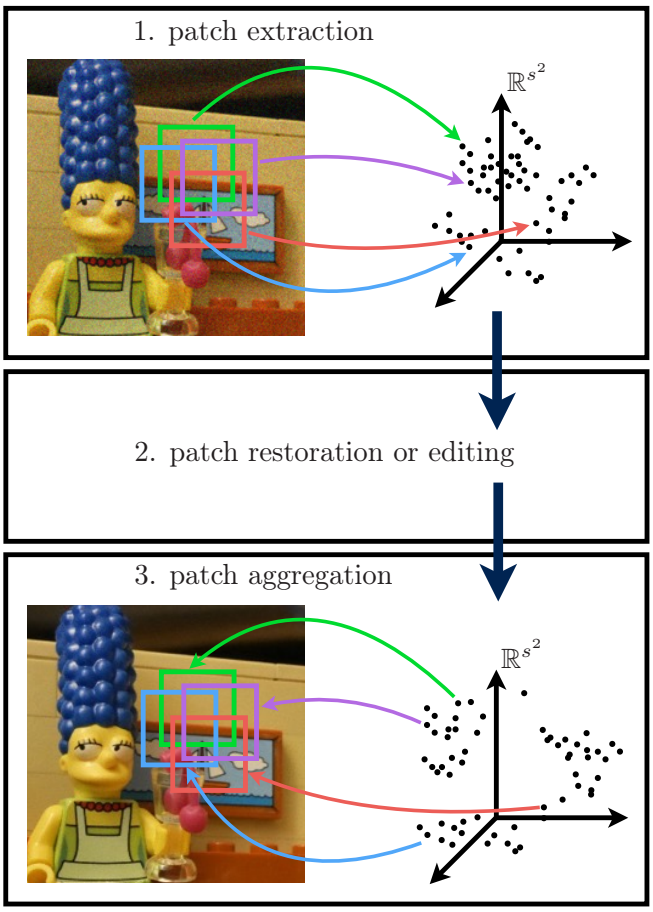


Fig. 1: Illustration of the 3 steps of patch-based image processing. Patches are small image pieces, they can be seen as vectors of a high dimensional space. Patch-based methods decompose images in overlapping patches (step 1) and make these patches collaborate for restoration, synthesis or editing purposes (step 2). At this point, the processed overlapping patches do not necessarily share the same values on their common pixels. Aggregation techniques aim at combining all these different overlapping patches into a single image (step 3).

However, after this processing, the set of patches is usually not in $\mathrm{Im}(E)$ anymore. It means that a pixel which belongs to several patches can have different values in all these processed patches.

For example, in image restoration, we have access to $\tilde{u}$, a distorted version of the true signal u. In order to construct $\hat{u}$, an estimate of u, we first extract the patches $(\tilde{y}_i)_{i \in [\![1,N]\!]} := E(\tilde{u})$ and we try to infer their restored versions $\hat{y}_i$. For instance, in a Bayesian framework, if we have access to a posterior probability distribution for each patch, we can estimate each $\hat{y}_i$ by

$$\hat{y}_i = \arg\max_y p(y|\tilde{y}_i).$$

However, after this estimation, there is no guaranty that $(\hat{y}_i)_{i\in[\![1,N]\!]} \in \mathrm{Im}(E)$, *i.e.* no guaranty that we can find $\widehat{u}$ such that $E(\widehat{u}) = (\hat{y}_i)_{i\in[\![1,N]\!]}$.

1.3 Patch aggregation

The patch aggregation is the action of recovering an image from a set of patches. It is characterized by an aggregation operator F, which reconstructs an image from a set of patches and must satisfy $F \circ E = \mathrm{Id}_{s_x \times s_y}$.

If E extracts all N overlapping square patches of size d from u, F can be seen as a map from $\mathbb{R}^{d\times N}$ to $\mathbb{R}^{s_x \times s_y}$. In this case, the most common aggregations are *the central pixel aggregation*, which consists in keeping only the central pixel of each patch and *the uniform (resp. weighted) aggregation* which consists in taking, at each pixel, the uniform (resp. weighted) average of the d different values provided by the patches it belongs to.

1.4 Aggregation of patch distributions

Patch aggregation is usually seen as a deterministic process. However, many patch-based methods rely on a probabilistic modeling, where a probability distribution is associated to each patch. In image restoration for instance, 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 (first line of Figure 2). We can argue that keeping a deterministic estimate for each patch seems quite reductive since no difference are made between distributions with small or large variances. In this context, it might be more interesting to reconstruct a probability distribution on the whole image from all these probability distributions on patches. The question of the aggregation of probability distributions is a natural generalization of the patch aggregation since (deterministic) patches can be seen as realizations of Dirac distributions. Solving this question enables to estimate directly a probability distribution on the whole image. This probability distribution can be used afterwards for estimation (second line of Figure 2). This is the path that we develop in the next Section.

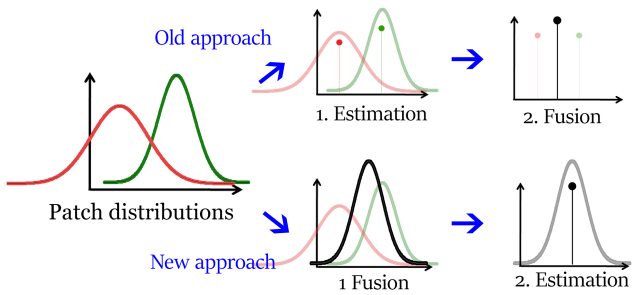


Fig. 2: Illustration of our new aggregation paradigm. On the first line, the classical process, which consists in computing estimates from patch distributions and then aggregating these estimates. On the second line, a simplified view of our framework, where we first aggregate probability distributions, and then use the fused model for estimation.

2 Patch model, agreement and fusion

In this Section, we define what we call a *patch model*, which extends the classical definition of a deterministic patch in a stochastic setting. This model will be used thereafter to define a notion of *patch fusion*, motivated by the situation described in Section 1.4.

2.1 Patch model: a probabilistic patch representation

Let us define a patch model of size d on the discrete grid $\mathbf{\Omega}$. The notion is illustrated by Figure 3.

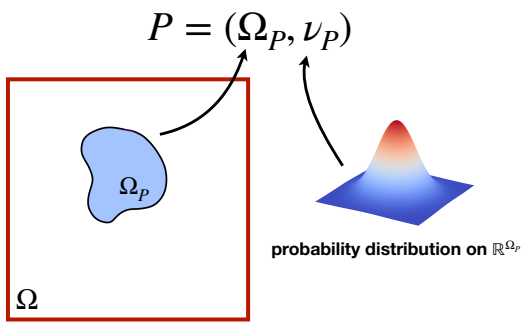
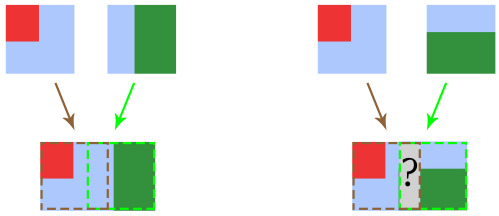


Fig. 3: A patch model on $\mathbf{\Omega}$ is composed of a domain Ω_P (a subset of $\mathbf{\Omega}$) and a probability distribution ν_P on $\mathbb{R}^{\Omega_P}$.

Definition 1 A patch model P on the grid $\mathbf{\Omega}$ is a couple (Ω_P, ν_P), where $\Omega_P \subset \mathbf{\Omega}$ and where 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 $\mathbf{\Omega}$.



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

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

Fig. 4: Illustration of the notion of agreement between two deterministic patches.

This definition is a generalization of the classical definition of a patch on a grid. Indeed, a deterministic patch P can be assimilated to a Dirac distribution on $\mathbb{R}^{\Omega_P}$. We do not impose any connectedness of Ω_P in our definition.

We are now in the position to define the notion of agreement between two patch models (see Figure 4), which says that two patch models agree if they share the same distribution on their intersection.

Definition 2 (Patch model agreement) Let $P_1 = (\Omega_1, \nu_1)$ and $P_2 = (\Omega_2, \nu_2)$ be two patch models in $\mathcal{P}$. We say that these two patch models agree and we write $P_1 \hat{=} P_2$ if and only if

$$\nu_{1|\Omega_1 \cap \Omega_2} = \nu_{2|\Omega_1 \cap \Omega_2}.$$

Therefore, two disjoint patch models (P_1 and P_2 such that $\Omega_1 \cap \Omega_2 = \emptyset$) agree automatically. The $\hat{=}$ relation is reflexive and symmetric, but not transitive. Observe that this definition can also be applied to deterministic patches. We say that they agree if their values on their overlap coincide. We will also denote this with the symbol $\hat{=}$.

We now define the notion of compatibility between patch models, which is much less restrictive that the patch agreement, and will be important to introduce the notion of patch fusion in the next Section.

Definition 3 (Patch model compatibility) Let $(P_n)_{n \in [\![1,N]\!]} = (\Omega_n, f_n(x)dx)_{n \in [\![1,N]\!]}$ be a set of N patch models with bounded densities $f_1, \ldots, f_n$. We say that these patch models are compatible if

$$\int_{z \in \mathbb{R}^{\cup_{n=1}^N \Omega_n}} \prod_{n=1}^{N} f_n(z_{|\Omega_n}) dz > 0.$$

2.2 Patch model fusion

We can now define the fusion of two patch models. As explained before, this definition is motivated by the situation where we end up with one or several distributions on the different patches. The fusion operation permits to construct directly a distribution for the whole image from the different patch models. It simply consists in aggregating patch models by merging their domains, and defining a novel distribution on this merged domain as a (specific) product of their original distributions.

Definition 4 (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 $P_1 \odot P_2$ is the patch model defined by (Ω, ν) where $\Omega = \Omega_1 \cup \Omega_2$ and $\nu(dx) = f(x)dx$, with

$$\forall x \in \mathbb{R}^{\Omega}, \quad f(x) = \frac{f_1(x_{|\Omega_1}) f_2(x_{|\Omega_2})}{\int_{z \in \mathbb{R}^{\Omega}} f_1(z_{|\Omega_1}) f_2(z_{|\Omega_2}) dz}.$$

Remark 1 – 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, and it is always satisfied with the distributions we consider, but it could be relaxed. In practice, we only need to ensure that

$$\int_{z \in \mathbb{R}^{\Omega}} f_1(z_{|\Omega_1}) f_2(z_{|\Omega_2}) dz < +\infty.$$

This fusion definition has a very intuitive motivation, as we shall see in the next proposition.

Proposition 1 (Interpretation of the fusion) *Let $P_1 = (\Omega_1, \nu_1)$ and $P_1 = (\Omega_1, \nu_2)$ be two compatible patch models and define $P_1 \odot P_1 = (\Omega, \nu)$. Assume that the distributions ν_1 and ν_2 have bounded densities f_1 and f_2. Let $Z_1 \sim \nu_1$ and $Z_2 \sim \nu_2$ be two independent random vectors. We write $Z_1 = \begin{pmatrix} X_1 \\ Y_1 \end{pmatrix}$, where X_1 corresponds to the coordinates of Z_1 on $\Omega_1 \cap \Omega_2$ (so $X_1 \sim (\nu_1)_{|\Omega_1 \cap \Omega_2}$) and Y_1 to the coordinates on $\Omega_1 \setminus \Omega_2$. We write $Z_2 = \begin{pmatrix} X_2 \\ Y_2 \end{pmatrix}$ in the same way, where X_2 corresponds to the coordinates of Z_2 on $\Omega_1 \cap \Omega_2$ and Y_2 on $\Omega_2 \setminus \Omega_1$ (Y_1 and Y_2 may not have the same dimension). Then ν is the conditional probability distribution of the vector $\begin{pmatrix} X_1 \\ Y_1 \\ Y_2 \end{pmatrix}$ given $X_1 = X_2$.*

Proof In the following, we denote by $p(X = x)$ the value of the density of the random variable X at x. For

$z = (x_1, y_1, y_2) \in \mathbb{R}^{\Omega_1 \cap \Omega_2} \times \mathbb{R}^{\Omega_1 \setminus \Omega_2} \times \mathbb{R}^{\Omega_2 \setminus \Omega_1}$, we want to calculate the conditional density

$$p\left(\begin{pmatrix} X_1 \\ Y_1 \\ Y_2 \end{pmatrix} = \begin{pmatrix} x_1 \\ y_1 \\ y_2 \end{pmatrix} | X_1 = X_2\right).$$

This conditional density can be written

$$\frac{p((X_1, Y_1, X_2, Y_2) = (x_1, y_1, x_1, y_2))}{p(X_1 - X_2 = 0)}$$

where

$$\begin{aligned} &p((X_1, Y_1, X_2, Y_2) = (x_1, y_1, x_1, y_2)) \\ &= p((X_1, Y_1) = (x_1, y_1)) \times p((X_2, Y_2) = (x_1, y_2)) \\ &= f_1\left(\begin{pmatrix} x_1 \\ y_1 \end{pmatrix}\right) \times f_2\left(\begin{pmatrix} x_1 \\ y_2 \end{pmatrix}\right) = f_1(z_{|\Omega_1}) \times f_2(z_{|\Omega_2}), \end{aligned}$$

by independence of Z_1 and Z_2. Moreover,

$$\begin{aligned} p(X_1 - X_2 = 0) &= \int p(X_1 = x_1, X_2 = x_1) dx_1 \\ &= \int p(X_1 = x_1) \times p(X_2 = x_1) dx_1. \end{aligned}$$

Since

$$p(X_1 = x_1) = \int f_1\left(\begin{pmatrix} x_1 \\ y_1 \end{pmatrix}\right) dy_1, \text{ and}$$

$$p(X_2 = x_1) = \int f_2\left(\begin{pmatrix} x_1 \\ y_2 \end{pmatrix}\right) dy_2, \text{ we conclude that}$$

$$p(X_1 - X_2 = 0) = \int f_1(z_{|\Omega_1}) \times f_2(z_{|\Omega_2}) dz > 0.$$

□

The fusion operation is therefore a way to combine two patch models while imposing these models to be equal on the intersection of their domains.

In order to extend this fusion operation to larger sets of patches, we need the following proposition.

Proposition 2 *For any compatible patch models with bounded densities P_1, P_2 and P_3 in $\mathcal{P}$, the fusion operation $\odot$ is well-defined and satisfies*

- *$P_1 \odot P_2 \in \mathcal{P}$ and has a bounded density.*
- *$(P_1 \odot P_2) \odot P_3 = P_1 \odot (P_2 \odot P_3)$.*
- *$P_1 \odot P_2 = P_2 \odot P_1$*

Proof Let $P_1 = (\Omega_1, f_1(x)dx)$, $P_2 = (\Omega_2, f_2(x)dx)$, $P_3 = (\Omega_3, f_3(x)dx)$ and $(\hat{\Omega}, \hat{f}(x)dx) = P_1 \odot P_2$. We have $\hat{\Omega} = \Omega_1 \cup \Omega_2$ and

$$\hat{f}(x) \propto f_1(x_{|\Omega_1}) \times f_2(x_{|\Omega_2}),$$

which clearly shows the commutativity. So $P_1 \odot P_2$ has also a bounded density and it is straightforward from the definition that $P_1 \odot P_2$ is compatible with P_3. Besides, if we have $(\bar{\Omega}, \bar{f}dx) = (P_1 \odot P_2) \odot P_3$, we get

$$\bar{f}(x) \propto f_1(x_{|\Omega_1}) \times f_2(x_{|\Omega_2}) \times f_3(x_{|\Omega_3}),$$

which clearly shows the associativity. □

Remark 2 This proposition ensures the stability and coherence of the operation, which can therefore be extended to any number of compatible patch models without ambiguity. For any set of compatible patch models with bounded densities written

$$(P_n)_{n \in [\![1,N]\!]} = (\Omega_n, f_n(x)dx)_{n \in [\![1,N]\!]},$$

we will denote this fusion by

$$\bigodot_n P_n = (\Omega, f(x)dx), \text{ with } \Omega = \bigcup_n \Omega_n \text{ and}$$

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

Merging patch models in any order will always yield the same result (under the condition of compatibility and bounded densities). This operation can be used to propagate and connect all patch models to obtain a single image model.

2.3 Fused image model

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

Definition 5 Let E be a set of patch models. We say that E *covers the image support* if every pixel of $\mathbf{\Omega}$ belongs to the domain of at least one patch model of E, i.e.

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

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.

For a set E of compatible patch models which covers the image support, Proposition 2 ensures that it is possible to fuse all the patch models of E to obtain a global model $(\mathbf{\Omega}, \nu) = \bigodot_{P \in E} P$ on the image. As a by-product, this constructs a new set $\hat{E}$ which represents the image, defined by

$$\hat{E} := \{(\Omega_P, \nu_{|\Omega_P}) \text{ with } P \in E\}.$$

Observe that the set $\hat{E}$ is generally different from the set E, even if E is obtained as all the marginals of a patch model on the whole image. Indeed, even if they agree, fusing two patch models does not preserve their common distribution on their intersection.

In practice, the previous definitions lead to generic algorithms which consist in fusing all patch models iteratively. This procedure is stable and justified by proposition 2, but not necessarily efficient. How the fusion is performed in practice should depend on the considered distributions.

In the case of normal or uniform patch models, we will see in the next section that the fusion has a closed-form solution. We did not investigate more involved models, but we think that approximate schemes could be used for more complex distributions.

3 Application to particular distributions

3.1 Uniform distribution

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

Proposition 3 (Fusion of uniform patch models) *Let $A \subset \mathbb{R}^{\Omega_A}$ and $B \subset \mathbb{R}^{\Omega_B}$ be two bounded borelian sets, 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 $C = \{x \in \mathbb{R}^{\Omega}; x_{|\Omega_A} \in A$ and $x_{|\Omega_B} \in B\}$.*

If $\gamma(C) > 0$, 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 $\mathbb{R}^{\Omega}$ whose marginals distributions on Ω_A and Ω_B are P_A and P_B.

This illustrates the behavior of the fusion operation, which forces patch models to agree on their intersection. As a consequence, a patch model with a peaked distribution will impose its opinion to the other patch models: we except a confident model to be given more credit in the final aggregation. As we shall see, the Gaussian case keeps this behavior, but in a softer way.

3.2 Gaussian distributions

The Gaussian distribution also yields a closed form expression for the fusion operation.

Proposition 4 (Fusion of Gaussian patch models) *Let $P_1 = (\Omega_1, \nu_1)$ and $P_2 = (\Omega_2, \nu_2)$ be two Gaussian patch models with positive definite covariances. We write x the variable representing the common pixels of the two patch models (*i.e. *those in $\Omega_1 \cap \Omega_2$) and y for the others (*i.e. *those in $\Omega_1 \setminus \Omega_2$ for P_1, and those in $\Omega_2 \setminus \Omega_1$ for P_2), and we write*

$$\nu_1 = \mathcal{N}\left(\begin{pmatrix}\mu_x\\ \mu_y\end{pmatrix}, \begin{pmatrix}\Sigma_x & \Sigma_{xy}\\ \Sigma_{xy}^T & \Sigma_y\end{pmatrix}\right)$$
$$\text{and } \nu_2 = \mathcal{N}\left(\begin{pmatrix}\mu'_x\\ \mu'_y\end{pmatrix}, \begin{pmatrix}\Sigma'_x & \Sigma'_{xy}\\ (\Sigma'_{xy})^T & \Sigma'_y\end{pmatrix}\right).$$

Then (Ω_1, ν_1) and (Ω_2, ν_2) are compatible and the distribution of $(\Omega_1, \nu_1) \odot (\Omega_2, \nu_2)$ is Gaussian with parameters

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

and

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

Proof Let $Z_1 = \begin{pmatrix}X_1\\ Y_1\end{pmatrix} \sim \nu_1$ and $Z_2 = \begin{pmatrix}X_2\\ Y_2\end{pmatrix} \sim \nu_2$ be two independent Gaussian random vectors. From proposition 1, we know that the distribution we are looking for is the conditional probability distribution of $\begin{pmatrix}X_1\\ Y_1\\ Y_2\end{pmatrix}$ knowing $X_1 = X_2$. The random variable $W = X_1 - X_2$ follows a Gaussian distribution with expectation $\mu_x - \mu'_x$ and covariance $\Sigma_x + \Sigma'_x$. Similarly, we know that $\begin{pmatrix}Z_1\\ Y_2\\ W\end{pmatrix}$ is a Gaussian random vector with parameters $\hat{\mu}, \hat{\Sigma}$ such that

$$\hat{\mu} = \begin{pmatrix}\mu_x\\ \mu_y\\ \mu'_y\\ \mu_x - \mu'_x\end{pmatrix} \text{ and } \hat{\Sigma} = \begin{pmatrix}\Sigma_x & \Sigma_{xy} & 0 & \Sigma_x\\ \Sigma_{xy}^T & \Sigma_y & 0 & \Sigma_{xy}^T\\ 0 & 0 & \Sigma'_y & -\Sigma'^T_{xy}\\ \Sigma_x^T & \Sigma_{xy} & -\Sigma'_{xy} & \Sigma_x + \Sigma'_x\end{pmatrix}.$$

Indeed, since $Z_1 = \begin{pmatrix}X_1\\ Y_1\end{pmatrix}$ and $Z_2 = \begin{pmatrix}X_2\\ Y_2\end{pmatrix}$ are independent, the covariance between Z_1 and W can be written

$$\mathrm{Cov}(Z_1, W) = \mathrm{Cov}(Z_1, X_1 - X_2) = \mathrm{Cov}(Z_1, X_1) = \begin{pmatrix}\Sigma_x\\ \Sigma_{xy}^T\end{pmatrix},$$

and the covariance between Y_2 and W is

$$\mathrm{Cov}(Y_2, W) = -\mathrm{Cov}(Y_2, X_2) = -\Sigma'^T_{xy}.$$

It follows that the conditional density of $\begin{pmatrix} X_1 \\ Y_1 \\ Y_2 \end{pmatrix}$ knowing $W = 0$ is a normal distribution with expectation

$$\begin{aligned}\boldsymbol{\mu} &= \mathbb{E}\begin{pmatrix} Z_1 \\ Y_2 \end{pmatrix} + \begin{pmatrix} \mathrm{Cov}(Z_1, W) \\ \mathrm{Cov}(Y_2, W) \end{pmatrix} \mathrm{Cov}(W, W)^{-1}(0 - \mathbb{E}(W)) \\ &= \begin{pmatrix} \mu_x \\ \mu_y \\ \mu'_y \end{pmatrix} + \begin{pmatrix} \Sigma_x \left(\Sigma_x + \Sigma'_x\right)^{-1} \\ \Sigma_{xy}^T \left(\Sigma_x + \Sigma'_x\right)^{-1} \\ -\Sigma_{xy}'^T \left(\Sigma_x + \Sigma'_x\right)^{-1} \end{pmatrix} (\mu'_x - \mu_x)\end{aligned}$$

and covariance matrix

$$\begin{aligned}\boldsymbol{\Sigma} &= \mathrm{Cov}\left(\begin{pmatrix} Z_1 \\ Y_2 \end{pmatrix}, \begin{pmatrix} Z_1 \\ Y_2 \end{pmatrix}\right) \\ &\quad - \begin{pmatrix} \mathrm{Cov}(Z_1, W) \\ \mathrm{Cov}(Y_2, W) \end{pmatrix} \mathrm{Cov}(W, W)^{-1} \begin{pmatrix} \mathrm{Cov}(Z_1, W)^T \\ \mathrm{Cov}(Y_2, W)^T \end{pmatrix} \\ &= \begin{pmatrix} \begin{pmatrix} \Sigma_x & \Sigma_{xy} \\ (\Sigma_{xy})^T & \Sigma'_y \end{pmatrix} & 0 \\ 0 & \Sigma'_y \end{pmatrix} - \begin{pmatrix} \Sigma_x \left(\Sigma_x + \Sigma'_x\right)^{-1} \\ (\Sigma_{xy})^T \left(\Sigma_x + \Sigma'_x\right)^{-1} \\ -(\Sigma'_{xy})^T \left(\Sigma_x + \Sigma'_x\right)^{-1} \end{pmatrix} \begin{pmatrix} \Sigma_x \\ \Sigma_{xy} \\ -\Sigma'_{xy} \end{pmatrix}^T\end{aligned}$$

□

Remark 3 We have defined the fusion only for distributions with densities, but in this case we see that we could extend the fusion to singular Gaussian distributions such that $\Sigma_x + \Sigma'_x$ is invertible.

As a consequence, 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.

If we merge N Gaussian patch models $(\Omega_n, \nu_n)_{n \in [\![1,N]\!]}$ with expectations $(\mu_n)_{n \in [\![1,N]\!]}$ and precision matrices [2] $(\Lambda_n)_{n \in [\![1,N]\!]}$, a very simple formula can be derived to link the parameters (μ, Λ) of the fused Gaussian model and the set $(\mu_n, \Lambda_n)_{n \in [\![1,N]\!]}$. Before giving this formula in the next proposition, note that we see the expectations μ_n as vectors of $\mathbb{R}^\Omega$ and matrices Λ_n as matrices of $\mathbb{R}^{\Omega \times \Omega}$, which means that $\mu_n(i)$ is the expectation of the patch n at the pixel i, and is thus defined only if i belongs to Ω_n.

Proposition 5 *Let $(\Omega_n, \nu_n)_{n \in [\![1,N]\!]}$ be N Gaussian patch models, with expectations $(\mu_n)_{n \in [\![1,N]\!]}$ and precision matrices $(\Lambda_n)_{n \in [\![1,N]\!]}$. Let $P = (\Omega, \nu) = \bigodot_{n \in [\![1,N]\!]} (\Omega_n, \nu_n)$. Then P is a Gaussian patch model, whose precision matrix Λ and expectation μ satisfy, for all $(i, j) \in \Omega \times \Omega$,*

$$\begin{aligned}\Lambda(i,j) &= \sum_{1 \le n \le N, i \in \Omega_n, j \in \Omega_n} \Lambda_n(i,j). \\ (\Lambda\mu)(i) &= \sum_{1 \le n \le N, i \in \Omega_n} (\Lambda_n \mu_n)(i).\end{aligned} \tag{1}$$

[2] The precision matrix is the inverse of the covariance matrix.

Proof We know that $\nu(dx) = f(x)dx$ is Gaussian and we have, according to remark 2,

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

Equations (1) follow by keeping for each pair (i, j) the indices n such that pixels i and j belongs to Ω_n.

□

Remark 4 Observe that while the precision matrices can be easily derived, the value of the expectation of the whole Gaussian is not directly accessible from (1), since the precision matrix needs to be inverted.

3.3 Fusion algorithm for Gaussian distributions

From the previous results, we can derive a simple and explicit fusion algorithm for normally distributed patches. In practice, if we aim at merging a set E of Gaussian patch models covering an image, keeping in memory and computing the covariance of the global Gaussian model is not tractable, since it requires to deal with a $(s_x \times s_y)^2$ matrix. Thanks to Proposition 5, we know that the precision matrix is sparse, but we have no such result on the covariance 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, as illustrated by Figure 5. It means that the covariance matrix of the whole image is almost sparse. This gives us the possibility to compute and store this covariance matrix much more easily, as described in algorithm 1.

In practice, this algorithm permits to compute the whole fused model. The computation is however quite slow: for a 512×512 image and 10×10 patches, fusing all patch models takes several minutes on a recent computer.

4 Link with classical aggregation methods

4.1 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.

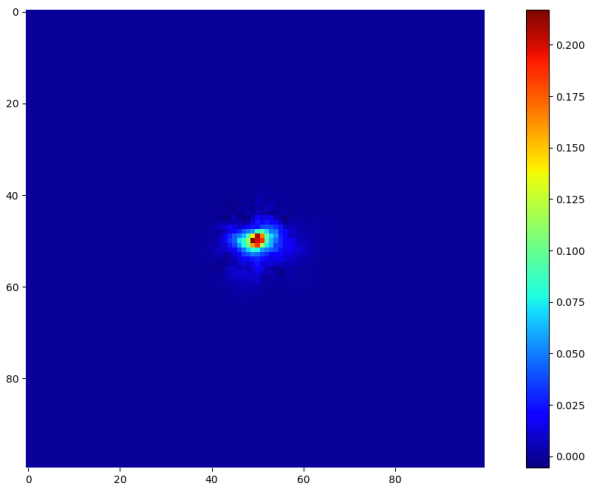


Fig. 5: In this experiment, we compute a complete Gaussian model for the image *Lena* thanks to our fusion algorithm. The figure shows the resulting (absolute values of the) correlation map of a pixel in a 100×100 patch. As we can see, the correlation decreases to 0 very fast when we move away from the center. This experiment was made using 10×10 patches, which justifies to approximate by 0 the correlation between pixels at distance greater than $2\sqrt{d}$.

Algorithm 1 Approximation of the fusion procedure for Gaussian models with sparsity hypotheses on the covariance matrix

Input: Set $\mathcal{P}$ of square patches of size d, block-size b
Output: Aggregated image $\boldsymbol{u}$

1: Compute $\mathcal{B}$, partition of the image domain composed of disjoint blocks of size $b \times b$.
2: $s \leftarrow 2 \times \sqrt{d}$
3: **for** $B \in \mathcal{B}$ **do**
4: $\quad \widetilde{B} \leftarrow$ block of size $(b+s) \times (b+s)$ centered in B
5: $\quad \mathcal{P}_{\widetilde{B}} \leftarrow \{P \in \mathcal{P} | P \subset \widetilde{B}\}$
6: $\quad$ Compute $u_{\widetilde{B}}$ by fusing iteratively all patches from $\mathcal{P}_{\widetilde{B}}$ using proposition 4
7: $\quad \boldsymbol{u}_{|B} \leftarrow u_{\widetilde{B}|_B}$
8: **end for**

Proposition 6 *Let (Ω_1, ν_1) and (Ω_2, ν_2) be two Gaussian patch models with diagonal positive definite covariances*

$$\nu_1 = \mathcal{N}\left(\begin{pmatrix}\mu_x \\ \mu_y\end{pmatrix}, \begin{pmatrix}\Sigma_x & 0 \\ 0 & \Sigma_y\end{pmatrix}\right)$$
$$\text{and } \nu_2 = \mathcal{N}\left(\begin{pmatrix}\mu'_x \\ \mu'_y\end{pmatrix}, \begin{pmatrix}\Sigma'_x & 0 \\ 0 & \Sigma'_y\end{pmatrix}\right),$$

where 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_1, ν_1) and (Ω_2, ν_2) are compatible and the distribution of $(\Omega_1, \nu_1) \odot (\Omega_2, \nu_2)$ is a Gaussian distribution with parameters

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

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

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

Proof This proposition is a direct application of proposition 4. □

The previous proposition states that if covariance matrices are all supposed diagonal, then the resulting fused 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 expectations of the patches containing this pixel. The weights are given by the precisions of the marginals at these pixel. We recognize here a special case of the *weighted aggregation procedure* described in the introduction. The more precise an estimate is, the more it counts in the final estimate.

A more specific case is the one obtained when all covariance matrices 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 expectation at a given pixel will be a simple average of all the expectations of the patches containing this pixel. This corresponds to the widely used *uniform aggregation*.

Finally, the limit case where each patch model has a covariance with infinite values except for its central pixel corresponds to the *central aggregation*.

4.2 Expected Patch Log Likelihood

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 [40]. Starting from an image

$$\tilde{u} = Au + \epsilon, \tag{2}$$

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 of u as one of the solutions of

$$\arg\min_u \frac{\lambda}{2\sigma^2}||Au - \tilde{u}||^2 - EPLL_f(u), \tag{3}$$

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

The authors of [40] interpret the quantity $EPLL_f(u)$ 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, whose proof follows directly from Remark 2.

Proposition 7 *Let u be an image on the domain $\mathbf{\Omega}$ and assume that $\nu(dx) = f(x)dx$ is a prior on all square patches of size $\sqrt{d} \times \sqrt{d}$ with domain inside $\mathbf{\Omega}$. Define E the set of all of of these square patch models sharing the same distribution $f(x)dx$. Then if $\bar{P} = \bigodot_{P \in E} P = (\bar{\Omega}, \bar{f}(x)dx)$ is well defined, there is a constant c such that*

$$EPLL_f(u) = \log \bar{f}(u) + c.$$

The function $EPLL_f$ is the log of the density obtained by fusing 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 $p(u)$ on the whole image u. Consequently, by choosing $\lambda = 1$ in equation (3), the solution of (3) can be interpreted as a maximum a posteriori and be written $\text{argmax}_u \log p(u|\tilde{u})$ on the whole image, since the term $-\frac{\lambda}{2\sigma^2}||Au - \tilde{u}||^2$ is, up to a constant, equal to $\log p(\tilde{u}|u)$ under the white Gaussian noise assumption.

Propositions 1 and 7 also clarify the link between the $EPLL$ approach and the iterative conditioning strategies used for instance in [28] for texture synthesis. Indeed, the fused image prior used in $EPLL$ can be interpreted as a probability distribution of a global random image obtained by fusing all patch distributions, conditioning by their equality on all their intersections.

Now, consider the pure denoising case ($A = Id$). In this case, the solution of (3) can also be interpreted as a maximum likelihood for another fused distribution $\bar{\tilde{f}}(x)dx$ on the whole image, as shown in the following distribution.

Proposition 8 *Keeping the notations of proposition 7, let $\bar{P} = \bigodot_{P \in E} P$ be the image model obtained by fusing all patch models of E. Let $P_{\tilde{u}} = \left(\mathbf{\Omega}, \mathcal{N}(\tilde{u}, \frac{\sigma^2}{\lambda} Id)\right)$ be an image model on the whole grid.*

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

$$\arg\min_u \frac{\lambda}{2\sigma^2}||u - \tilde{u}||^2 - EPLL_f(u) = \arg\max_u \bar{\tilde{f}}(u).$$

Proof We just have to remark that

$$\log \bar{\tilde{f}}(u) = \log \bar{f}(u) + \log\left(e^{-\lambda \frac{\|u-\tilde{u}\|^2}{2\sigma^2}}\right) + cst$$
$$= EPLL_f(u) - \frac{\lambda}{2\sigma^2}||u - \tilde{u}||^2 + cst. \square$$

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 $\frac{\sigma^2}{\lambda}$.

Under the full degradation model (2), a last interpretation of (3) 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

$$\tilde{u}_{|\Omega} = A_{|\Omega} u_{|\Omega} + \epsilon_{|\Omega}.$$

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 $f(x)dx$. For the sake of simplicity, we assume in the following proposition that each pixel is covered by exactly the same number of patch models. This is true if we assume that the image is periodic. In practice, it is satisfied for all pixels except those lying close to the image borders.

Proposition 9 *Keeping the notations of proposition 7, assume that each pixel of $\mathbf{\Omega}$ is covered by exactly d 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*

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

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

$$\log \bar{f}_{ap}(u) = EPLL_f(u) - \frac{d}{2\sigma^2}||Au - \tilde{u}||^2 + cst.$$

Proof We just have to remark that

$$\sum_{P \in E} \|A_{|\Omega} u_{|\Omega} - \tilde{u}_{|\Omega}\|^2 = d||Au - \tilde{u}||^2.$$

$\square$

In other words, for $\lambda = d$, the solution of (3) is a maximum of a fused posterior model on the whole image, assuming that all patches have the same prior $f(x)dx$.

4.3 Bayesian Model Averaging

We can ask ourselves the question of the link between the fusion operation introduced in this article and the notion of Bayesian Model Averaging (BMA) [18], which also attempts to combine information provided by different models on data. For the sake of simplicity, assume that we have two patches models P_1 and P_2 on the same domain Ω, and an observed degraded patch y on Ω. In the BMA framework, the a posteriori distribution of the (unknown) clean patch x can be written

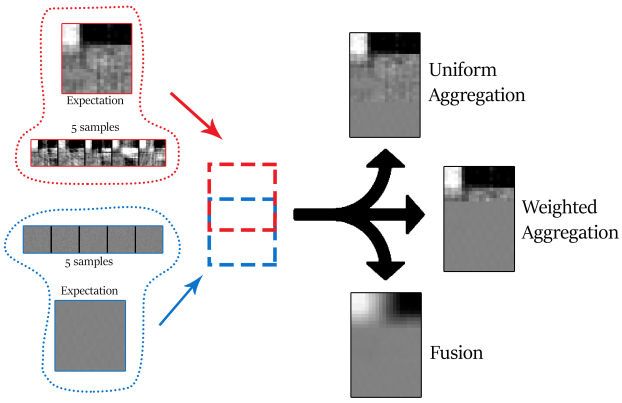
$$p(x|y) = \sum_{k=1,2} p(x|y, P_k)p(P_k|y),$$

where each $p(x|y, P_k)$ is simply the a posteriori distribution of x knowing y for the patch model P_k. Since $p(P_k|y)$ is a scalar, the BMA of two posterior models is merely a linear combination of these models. It can be interpreted as a generalization of the weighted aggregation, but is different from the fusion operation.

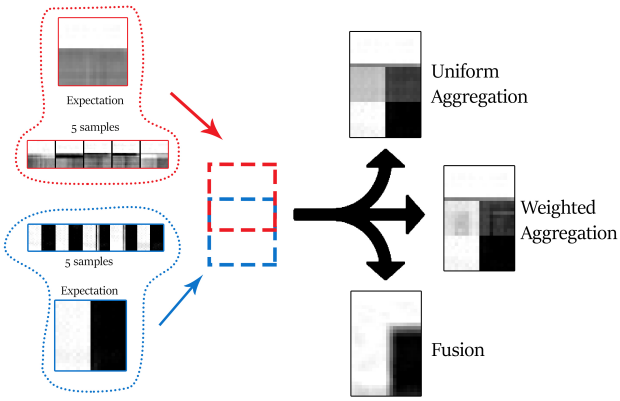
5 Experiments

In this Section, we illustrate the behavior of the fusion operation on different examples. We start with toy examples showing the main difference between the fusion and the classical uniform and weighted aggregations. Then we focus on the particular case of patch-based image denoising and we give some insight on the advantages and limitations of the fusion. We conclude with two simple extensions: the first one consists in mixing the fusion and the uniform aggregation in order to keep the best of both worlds, and the second one consists in a sparse fusion relying on a very few number of patch models.

Throughout this experimental section, we focus on the case of Gaussian distributions and we compute the expectations and covariances of the fused models explicitly, as explained in Section 3.3. However, let us underline that this is usually not the most efficient way to take advantage of the fused model. Indeed, we have seen in Remark 2 that the logarithm of the fused density can be written directly by summing the logarithms of these densities. As a consequence, it is very easy to integrate such a model in any variational framework, without any explicit computation of the fused model, even if this won't necessary yield a convex formulation (it will be convex for normal densities though).



(a) In this case, the two patch models almost agree, but the red one has a very large variance compared to the blue one.



(b) In this case, the two patch models completely disagree.

Fig. 6: Illustration of the behavior of the different aggregation schemes for two adjacent Gaussian patch models. On both figures, the two patch models on the left are aggregated in three different ways to form the patches on the right, either with the uniform aggregation, the weighted aggregation (taking into account the precision of each pixel) and the fusion operation.

5.1 A toy example

Figure 6 shows two toy examples clarifying the difference between the fusion defined in this paper on the one hand, and the weighted and uniform aggregation on the other hand. In these examples (a) and (b), two Gaussian patch models (shown respectively in red and blue on the left) are fused and the three aggregation strategies lead to quite different results. In both examples, the red model has a high variance and the blue model is more precise (or more confident, if we see patch models as persons with more or less solid opinions). On the right, we show only the expectations of the fused models.

Fig. 7: Set of images used to illustrate the results. From left to right: *Lena*, *Barbara*, *cartoon* and *squares*.

In both examples, the uniform aggregation gives the same credit to both patch models, whatever their covariances. In (a), although both patch models seem to almost agree on their overlap, this results in a quite noisy result on the patches overlap, even if the blue model has a very high precision in this region. The weighted aggregation takes into account this precision and yields a more satisfying result. The fusion operation also gives much more credit to the blue patch model than to the blue one and yields a much smoother result. As we shall see in the section devoted to image denoising, this behavior permits to obtain very regular results, free from the usual artifacts created by standard aggregation procedures, but at the price of some blur.

In example (b), both patch models strongly disagree on their overlap. The uniform aggregation yields a result which can be seen as a compromise between their opinions but is in contradiction with both of them. The weighted aggregation takes into account the greater precision of the blue model but still yields a result which is quite unlikely from both models point of views. Again the fusion yields a quite smooth result, which is likely for both patch models (even if it is more likely for the blue model than for the red one).

Notice that the uniform and weighted aggregations do not update the pixels outside of the patch overlap zone, while the fusion operation also affects these pixels, as shown in Proposition 4.

5.2 Application to denoising

For the sake of simplicity, we restrict the rest of 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 $\mathbf{\Omega}$.

We recall here the patch-based restoration framework applied to denoising. In image denoising, in order to restore an unknown image u from its noisy version $u + \varepsilon$, we usually start by extracting all square patches $\{y_k,\ k \in \{1, \dots, |\mathbf{\Omega}|\}\}$ from $\tilde{u} = u + \epsilon$. The noise model on patches can be written

$$y_k = x_k + \epsilon_k,$$

NL-Bayes				
Aggregation	*Uniform*	*Weighted*	*Fusion* $\lambda = 1$	*Fusion* $\lambda = 10$
Lena	30.58	30.49	30.28	**30.66**
Barbara	28.99	28.94	28.83	**29.04**
Cartoon	30.04	29.98	29.57	**30.35**
Squares	45.28	46.87	**47.35**	46.54

EPLL Model				
Aggregation	*Uniform*	*Weighted*	*Fusion* $\lambda = 1$	*Half Quadratic Splitting*
Lena	30.69	30.42	29.88	**30.71**
Barbara	26.56	26.18	25.45	**27.55**
Cartoon	29.89	29.62	28.65	**30.49**
Squares	37.38	39.09	36.96	**39.51**

HDMI				
Aggregation	*Uniform*	*Weighted*	*Fusion* $\lambda = 1$	*Fusion* $\lambda = 10$
Lena	**31.12**	31.10	28.16	29.96
Barbara	**29.55**	29.54	25.57	28.72
Cartoon	**30.55**	30.52	25.67	29.34
Squares	44.24	**48.77**	46.37	35.62

Fig. 8: PSNR of the different aggregation methods with NL-Bayes inference.

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 u from $u+\epsilon$:

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 $\hat{u}$ 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 sections, we will focus on three of them, which are representative of different choices in the three previously mentioned steps:

- NL-Bayes [23], which estimates a specific Gaussian model $\mathcal{N}(\mu_k, \Sigma_k)$ for each patch x_k;
- HDMI [19], which estimates a low-dimensional Gaussian Mixture model for the whole set of patches $x_k,\ k \in \{1, \dots, |\mathbf{\Omega}|\}$;
- EPLL [40], which estimates a Gaussian Mixture model for patches on an external database, and replaces

steps 2 and 3 above by the variational problem (3) and solves it by Half Quadratic Splitting.

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 component 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 fused using the framework introduced in the previous Sections. If we compute a fused 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 fused image model density, the restored image is computed as the solution of

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

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

$$\text{argmax}_u \bar{f}(u|\tilde{u}).$$

Now, writing x_k for the patches of u,

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

Thus, both strategies boil down to minimize an energy of the same form

$$\text{argmin}_u \frac{\lambda}{2\sigma^2}\|u-\tilde{u}\|^2 - \log \bar{f}(u), \tag{5}$$

with different values of λ. The value $\lambda = 0$ corresponds to the fusion of the prior models and $\lambda = d$ corresponds to the fusion of the posterior models. Fusing with higher values of λ gives much more weight to the noisy image $\tilde{u}$.

5.3 Results for the three different inference methods

Experiments are led on four different 512×512 images, *Lena*, *Barbara*, *Cartoon* and *Squares*. We will see that the behaviors of the different aggregation procedures strongly depend both on the image content and on the way patch Gaussian models are inferred from the noisy data.

For each of the three inference methods described in the previous section (NLBayes [23], HDMI [19] and EPLL [40], we provide different visual results illustrating the visual effects of all aggregation strategies. PSNR values are also provided in Table 8. As we shall see, while the complete fusion operation is not really competitive PSNR-wise, it leads interesting visual results, quite different from the more simple aggregation strategies. Our goal here is to study and highlight these different behaviors.

5.3.1 NL-Bayes

The NL-Bayes algorithm infers a specific Gaussian models for each patch and uses very small patches (5×5). As a consequence, most Gaussian covariances are quite well approximated by their diagonal, and the different aggregation procedures only display minor differences on natural images. Table 8 shows that the fusion slightly improves the PSNR results, but the difference is more significant for very simple geometric images like *squares*, even if the visual differences are quite subtle and concentrated around the junctions and edges of the rectangles.

5.3.2 EPLL

The EPLL model [40] makes use of 8×8 patches and learns a Gaussian mixture model with 200 groups on a large external set of images. In the original paper, patches are centered (their DC component is removed) before processing and all the Gaussian models from the GMM are assumed to have zero means. Additionally, in order to minimize (3), the authors of [40] introduce an auxiliary variable and make use of Half-Quadratic Splitting, which means that the restoration iterates between patch estimation and image reconstruction (by a uniform aggregation). In consequence, while the $EPLL_f$ term is a particular case of the fusion operation, their model is not directly comparable to our framework.

First, in order to use the closed-form expressions of Proposition 4, we need a Gaussian model for each patch and not a full GMM. As explained above, we chose to keep as a prior for each patch the Gaussian of the mixture which is the more likely for it. However,

Fig. 9: Comparison of the different aggregation procedures on *Lena* and *Barbara* using EPLL for the inferrence of Gaussian models. Images are degraded by a noise of standard deviation $\sigma = 30$.

observe that by making this choice we loose some of the information of the full GMM model. Second, we need a model on patches, and not on centered patches. To cope with this limitation, we remove the mean value of each patch, select the most likely Gaussian component in the GMM, and give the mean value of the original patch to this Gaussian model. As a result, the expectations of the different Gaussian models contain an important low frequency noise. For these different reasons, the comparison of the different aggregation strategies with the EPLL algorithm (which makes the fusion on the full GMM model) should be made with caution. Figure 9 provides the result of these different strategies for the images *Lena* and *Barbara* with $\sigma = 30$. The results of the fusion operation on these models are very smooth but present what we call a "fluffy" effect (see Section 5.4.1 and Figure 11), due to the way the Gaussian means are handled.

5.3.3 HDMI

In the HDMI algorithm [19], a GMM is learned on 10×10 patches, with only a few dozens of low-dimensional Gaussians in the mixture. Again, we keep as a prior for each patch the Gaussian of the mixture which is the more likely for it, so we loose part of the richness of the original model in our experiments. Figure 10 provides the result of the different strategies for the images *Lena* and *Barbara* with $\sigma = 30$. In this case, the different aggregations procedures produce quite important differences. The uniform aggregation is efficient PSNR-wise, but suffers from numerous artifacts (see Section 5.4.2). Using the whole fused model provides results which are below PSNR-wise but are also much smoother, and removes numerous artifacts. Some of the results suffer from a loss of contrast, explained in Section 5.4.3.

It is noticeable that the fusion operation tends to improve the results for the models inferred by NL-Bayes while it does not for the HDMI and EPLL models, at least PSNR-wise. We think that it can be partly ex-

Fig. 10: Comparison of the different aggregation procedures on *Lena* and *Barbara*, using HDMI for the inferrence of Gaussian models. Images are degraded by a noise of standard deviation $\sigma = 30$.

plained by the fact that at this point, we are able to take fully into account the Gaussian models inferred by NL-Bayes and that it is not the case for the GMM in HDMI and EPLL.

5.4 Visual effects

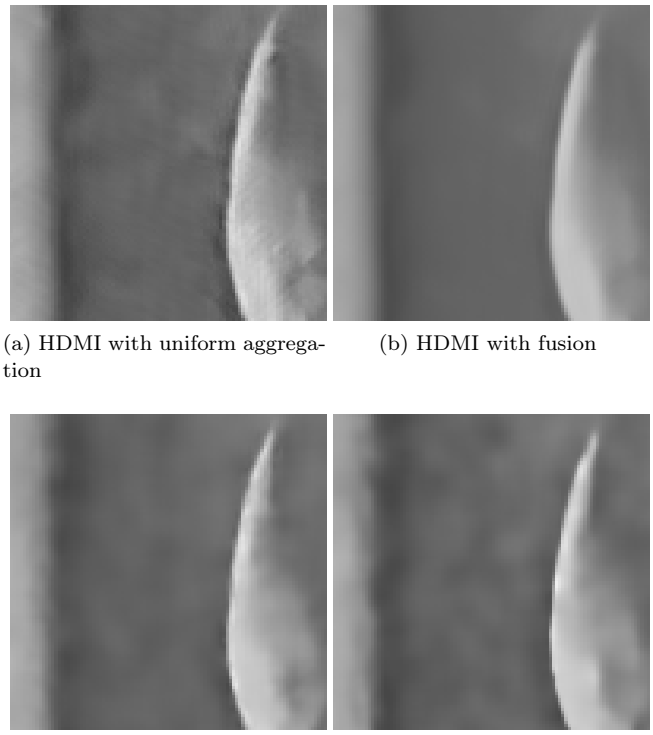
5.4.1 Fluffy effect

We call "fluffy effect" the effect visually similar to cotton, appearing in constant regions when using single scale patch-based methods (see Figure 11).

When using HDMI [19], the fusion aggregation clearly reduces this defect, whereas it does increase when using EPLL [40]. We can explain these results as follows: in HDMI, patch priors have (almost) independent expectations, since they are inferred using numerous different patches on the whole image. The remaining inconsistencies between overlapping patch models are thus removed by the fusion. With EPLL, since each noisy patch has its own DC component as a model expectation, and since these DC components are not independent for overlapping patches, the white noise low frequencies are reinforced by the fusion and the results show a very pronounced fluffy effect.

5.4.2 Artifacts

The main advantage of the fusion aggregation is to reduce the artifacts. This is quite understandable, since the method creates a model for which all overlapping patches have to agree. An artifact is created when one or several of the original overlapping patch models are badly estimated. In this case, even if the uniform aggregation averages several correct estimates with this wrong estimate, the artifact can remain noticeable. When using the fusion approach, if this artifact is inconsistent with the other models, it will completely disappear. This is illustrated by Figure 12.



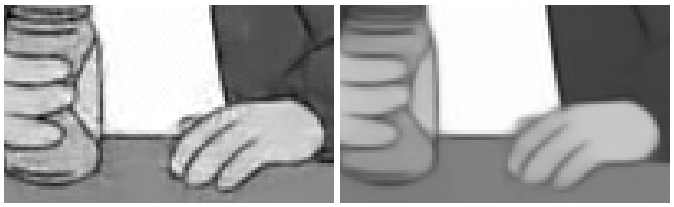
(a) HDMI with uniform aggregation (b) HDMI with fusion

(c) EPLL model with uniform aggregation (d) EPLL model with fusion

Fig. 11: Influence of the fusion on the fluffy effect (low frequency noise visually similar to cotton and visible in constant regions after patch-based processing). On the first line, when using the HDMI algorithm, the fluffy effect is highly reduced by the fusion, since the hypothesis that the patch models are independent is almost fulfilled. On the contrary, when using EPLL, the average of the noisy patch becomes the expectation of its prior model. In this case, the fluffy effect is amplified by the fusion.

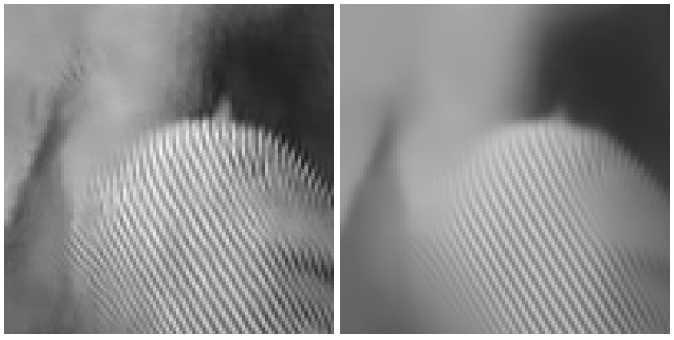
5.4.3 Blur and constrast

The main drawback of the fusion operation is a loss of constrast and sharpness around some geometric structures, which makes the PSNR decrease. This is particularly striking in regions where the patch models are not well learned. In practice, 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, and will result in an important contrast loss. 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. Besides, if a texture or an edge is not captured by the model, then the fusion cannot restore it properly and the resulting texture will appear blurry. This effect is illustrated on Figure 13 and can be reduced for instance by using the information of the fused model, see Section 5.5.1.



(a) HDMI with uniform aggregation (b) HDMI with fusion

Fig. 12: Examples of artifacts created in patch-based image denoising. On the left, we can see that the uniform aggregation creates numerous artifacts, for instance around the fingers. These artifacts, which are inconsistent across overlapping patch models, are not present in the fusion result.



(a) HDMI with uniform aggregation (b) HDMI with fusion

Fig. 13: Illustration of the loss of sharpness and constrast due to the fusion operation. As we can see, the stripes of the legs of Barbara are perfectly restored by the fusion aggregation, since the model is well-trained on this region. However, on the sides of the leg, the texture looks blurry. This is explained by the lack of patch examples for this geometry. The bound of the shadow on the arm is also well-restored by the fusion, but the "dark spot" on the elbow is another good example of contrast loss: many patch models of this region are considered to be uniform and highly reduce the obscurity of the area.

5.5 Possible extensions

5.5.1 Precision aggregation

As we have seen, the fusion yields good results in regions where the estimated model is confident and has been well trained. This "confidence" level can be accessed through the covariance of the fused model. A simple way to exploit it is to consider the precision of the marginal at a given pixel. If this precision is high, we can consider to keep the estimate provided by the fusion, and use another estimate otherwise, like the uniform aggregation. This way, we can construct the *precision estimate*, defined as an average of the images obtained by the uniform aggregation and the fusion, weighted by the precision of the marginals for each pixel. This idea is illustrated on Figure 14. The figure shows the precision map obtained on *Lena* with the Gaussian models of the HDMI algorithm, and the resulting *precision aggregation*, which clearly keeps the best of both worlds, reducing the artifacts of the uniform aggregation but providing a much less blurry result than the sole fusion.

5.5.2 Sparse aggregation

The fusion does not need numerous patches at each pixel to achieve visually smooth results. The image can therefore be reconstructed using a reduced number of patches, chosen either at random or using some heuristics to select the best model among them, as in [33] for instance. This could be a way to speed up the learning phase, or to spend more time learning more complicated models. Figure 15 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.

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 built on this notion to propose a general common framework for the aggregation operation, seen as a fusion of different overlapping patch models. As we have shown, this common framework includes all previous aggregation schemes used in the literature, and reduces the design of new ones to the design of a fusion operation.

Our approach also permits to compute a fused image model which generalizes the Expected Patch Log Likelihood introduced by [40]. When patches are assumed to follow Gaussian distributions, this fused model is also Gaussian, with tractable expectation and covariance. This whole fused 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 fusion to aggregate does not necessarily improve the result PSNR-wise, but highly reduces the visual artifacts. On some images, it sometimes outperforms the standard uniform and weighted aggregations. The fusion is preferable if the model is well trained, since it takes advantage of all the provided information. However, patch models are in practice difficult to infer, in particular on natural images.

In the near future, we intend to apply this aggregation to other types of inverse problems, to wider classes of signals (for instance 1D signals from audio processing), or to other classes of models, like Gaussian mixtures. We also wish to investigate other kind of fusion operations. Indeed, the proposed fusion has some limitations. For instance, when merging two Gaussian patch models, one could expect that the resulting covariance would depend on the expectations of the original Gaussian models, but this is not the case with the proposed definition. It should be interesting to investigate more deeply in this direction.

Acknowledgment.

We would like to thank Arthur Leclaire and Pablo Arias for their insightful comments. We also would like to thank the three reviewers of this paper, whose comments and remarks have greatly helped to improve this work. 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.

(a) precision map (b) uniform aggregation (c) fusion (d) precision estimate, mix of (b) and (c)

Fig. 14: 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 confident the model is for each pixel. This enables to compute the precision estimate, which tries to keep the best of both worlds.

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. Diego Carrera, Giacomo Boracchi, Alessandro Foi, and Brendt Wohlberg. Sparse overcomplete denoising: aggregation versus global optimization. *IEEE Signal Processing Letters*, 24(10):1468–1472, 2017.
5. Taeg Sang Cho, Moshe Butman, Shai Avidan, and William T Freeman. The patch transform and its applications to image editing. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.
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. Aram Danielyan, Vladimir Katkovnik, and Karen Egiazarian. Bm3d frames and variational image deblurring. *IEEE Transactions on Image Processing*, 21(4):1715–1728, 2012.
8. 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.
9. Zhou Dengwen and Shen Xiaoliu. Image denoising using weighted averaging. In *Communications and Mobile Computing, 2009. CMC'09. WRI International Conference on*, volume 1, pages 400–403. IEEE, 2009.
10. 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.
11. 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.
12. 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.
13. Michael Elad and Peyman Milanfar. Style transfer via texture synthesis. *IEEE Trans. Image Processing*, 26(5):2338–2351, 2017.
14. Jianzhou Feng, Li Song, Xiaoming Huo, Xiaokang Yang, and Wenjun Zhang. An optimized pixel-wise weighting approach for patch-based image denoising. *IEEE Signal Processing Letters*, 22(1):115–119, 2015.
15. 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.
16. Onur G Guleryuz. Weighted averaging for denoising with overcomplete dictionaries. *IEEE Transactions on Image Processing*, 16(12):3020–3034, 2007.
17. 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.
18. Jennifer A Hoeting, David Madigan, Adrian E Raftery, and Chris T Volinsky. Bayesian model averaging: a tutorial. *Statistical science*, pages 382–

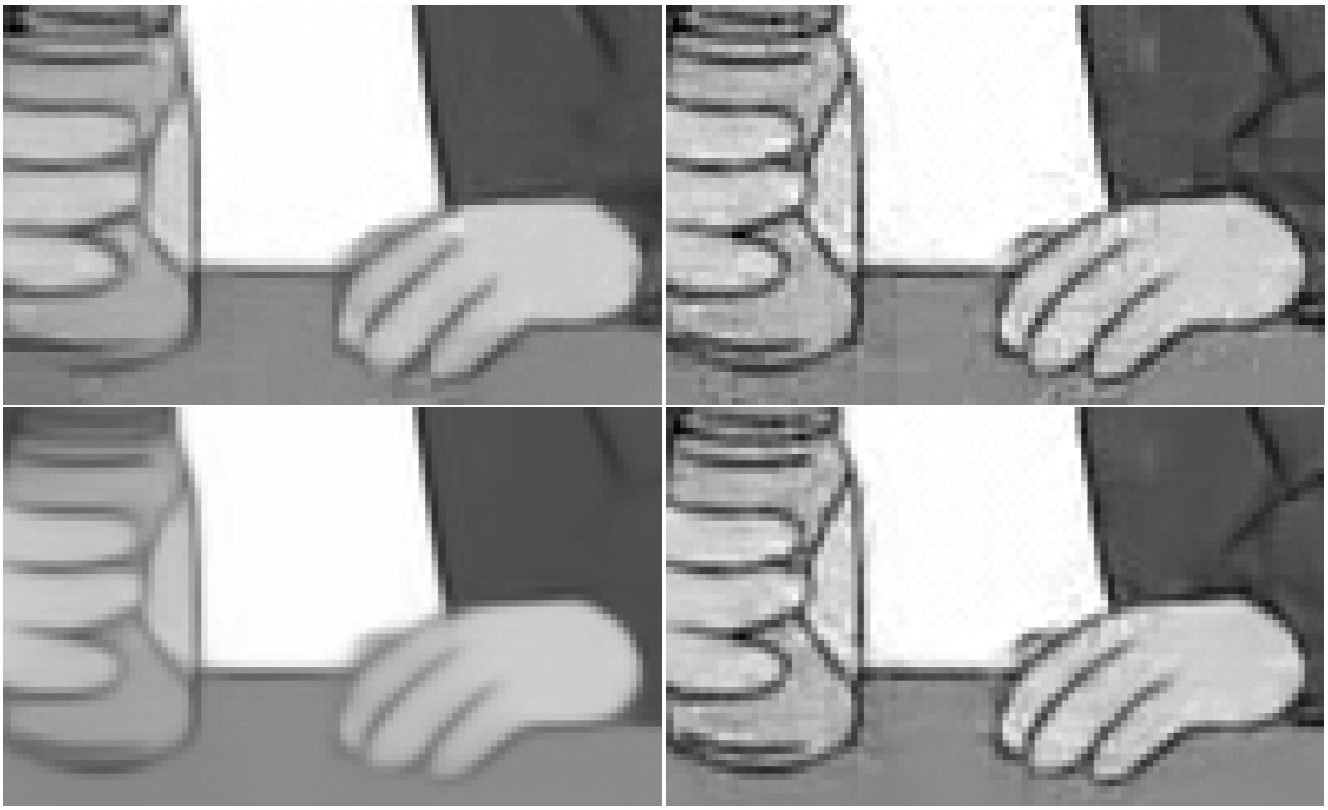


Fig. 15: Comparison of different aggregation procedures on the cartoon image. On the top row, only 4% of the patches are used, and on the bottom row, all the patches are used. On the left the fusion, and on the right the uniform aggregation

401, 1999.

19. Antoine Houdard, Charles Bouveyron, and Julie Delon. High-dimensional mixture models for unsupervised image denoising (hdmi). *SIAM Journal on Imaging Sciences*, 11(4):2815–2846, 2018.
20. Charles Kervrann. Pewa: Patch-based exponentially weighted aggregation for image denoising. In *Advances in Neural Information Processing Systems*, pages 2150–2158, 2014.
21. Charles Kervrann and Jérôme Boulanger. Optimal spatial adaptation for patch-based image denoising. *IEEE Transactions on Image Processing*, 15(10):2866–2878, 2006.
22. 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.
23. Marc Lebrun, Antoni Buades, and Jean-Michel Morel. A nonlocal bayesian image denoising algorithm. *SIAM Journal on Imaging Sciences*, 6(3):1665–1688, 2013.
24. Marc Lebrun, Miguel Colom, Antoni Buades, and Jean-Michel Morel. Secrets of image denoising cuisine. *Acta Numerica*, 21:475–576, 2012.
25. 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.
26. Ignacio Francisco Ramírez Paulino. Paco: Signal restoration via patch consensus. *arXiv preprint arXiv:1808.06942*, 2018.
27. Nicola Pierazzo, Jean-Michel Morel, and Gabriele Facciolo. Multi-scale dct denoising. *Image Processing On Line*, 7:288–308, 2017.
28. 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.
29. Yaniv Romano and Michael Elad. Boosting of image denoising algorithms. *SIAM Journal on Imaging Sciences*, 8(2):1187–1219, 2015.
30. Stefan Roth, Victor Lempitsky, and Carsten Rother. Discrete-continuous optimization for optical flow estimation. In *Statistical and Geometrical Approaches to Visual Motion Analysis*, pages 1–22. Springer, 2009.

31. 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.
32. OG Sezer and Y Altunbasak. Weighted average denoising with sparse orthonormal transforms. proceedings of the ieee international conference on image processing (icip). *Cairo, Egypt*, 2009.
33. 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.
34. Hossein Talebi, Xiang Zhu, and Peyman Milanfar. How to saif-ly boost denoising performance. *IEEE Transactions on image processing*, 22(4):1470–1485, 2013.
35. 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.
36. Dimitri Van De Ville and Michel Kocher. Sure-based non-local means. *IEEE Signal Process. Lett.*, 16(11):973–976, 2009.
37. Yi-Qing Wang and Jean-Michel Morel. Sure guided gaussian mixture image denoising. *SIAM Journal on Imaging Sciences*, 6(2):999–1034, 2013.
38. 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.
39. Maria Zontak, Inbar Mosseri, and Michal Irani. Separating signal from noise using patch recurrence across scales. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1195–1202, 2013.
40. 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.