This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2912122, IEEE Transactions on Image Processing

External Patch-Based Image Restoration Using Importance Sampling

Milad Niknejad, José Bioucas-Dias *Fellow, IEEE*, Mário A.T. Figueiredo *Fellow, IEEE*

***Abstract*—This paper introduces a new approach to patch-based image restoration based on external datasets and importance sampling. The *minimum mean squared error* (MMSE) estimate of the image patches, the computation of which requires solving a multidimensional (typically intractable) integral, is approximated using samples from an external dataset. The new method, which can be interpreted as a generalization of the external non-local means (NLM), uses self-normalized importance sampling to efficiently approximate the MMSE estimates. The use of self-normalized importance sampling endows the proposed method with great flexibility, namely regarding the statistical properties of the measurement noise. The effectiveness of the proposed method is shown in a series of experiments using both generic large-scale and class-specific external datasets.**

***Index Terms*—Image restoration, image denoising, patch-based methods, non-local means, minimum mean squared error, importance sampling.**

I. Introduction

Under the framework of imaging inverse problems, image restoration aims at reverting the degradation introduced by the image acquisition process. The observation is often modeled as a linear function of the underlying original image, contaminated by noise. Formally, with $\mathbf{x} \in \mathbb{R}^n$ being the vector whose entries are the (lexicographically ordered) pixel values of the original image, and $\mathbf{y} \in \mathbb{R}^m$ denoting the observed image, the linear model is written as

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{v}, \tag{1}$$

where $\mathbf{v}$ models additive noise (often zero-mean, independent, Gaussian distributed), and $\mathbf{H} \in \mathbb{R}^{m \times n}$ is a matrix modeling the observation/degradation process (blur, convolution, projection, etc...). To address the image restoration problem using Bayesian tools, a central building block is the translation of the forward model (1) into a conditional probability (density or mass) function of $\mathbf{y}$, given $\mathbf{x}$, also called the *likelihood function*. In the case of Gaussian noise, this yields a well-know conditional *probability density function* (pdf)

$$f_{Y|X}(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}|\mathbf{H}\mathbf{x}, \sigma^2\mathbf{I}) \propto \exp\Big(-\frac{1}{2\,\sigma^2}\|\mathbf{y} - \mathbf{H}\,\mathbf{x}\|_2^2\Big), \tag{2}$$

where σ^2 is the noise variance and $\mathbf{I}$ denotes an identity matrix.

In many important cases, the observation model departs from (1), namely because there are non-linear effects or the noise is not additive. For example, in a very large and important class of problems [1], [2], [3], the observations follow a Poisson distribution; formally, each of the observations, $(\mathbf{y})_l \in \mathbb{N}_0$, for $l \in 1, \ldots, m$, is a realization of a Poisson random variable with mean $(\mathbf{H}\mathbf{x})_l$. Under the standard conditional independence assumption, the conditional probability mass function of $\mathbf{y}$ given $\mathbf{x}$ is thus

$$f_{Y|X}(\mathbf{y}|\mathbf{x}) = \prod_{l=1}^{m} \frac{e^{-(\mathbf{H}\mathbf{x})_l}(\mathbf{H}\mathbf{x})_l^{(\mathbf{y})_l}}{(\mathbf{y})_l!}, \tag{3}$$

where a subscript $(\cdot)_i$ denotes the i-th component of the corresponding vector.

Image restoration methods are often specified for one type of observation and noise model. A significant amount of work has been devoted to Gaussian denoising, which corresponds to (1) with $\mathbf{H} = \mathbf{I}$. Some methods rely only on the noisy image itself [4], [5], [6], [7], [8], [9], on an external dataset of clean images [10], [11], or a combination of both [12], [13]. Methods based on external datasets have been recently an active research topic, especially with the emergence of *deep neural networks* (DNN), which require a large set of training data (see [14], [15], [16], [17], [18] and references therein). Most of those methods require separate training for different image restoration tasks and even for different parameters of the observation model, namely the noise variance. A different approach, also based on an external dataset of images, learns some parametric distribution for image patches, often *Gaussian mixture models* (GMMs) [11], [12]. The learned models are then used to as priors/regularizers in the image restoration process. However, those methods have been mostly limited to linear inverse problems with Gaussian noise, due to the difficulty in obtaining MAP or MMSE estimates for other noise models.

Another major group of image restoration methods based on external datasets works by using sample patches (instead of learning parametric models) to restore the observed patches [13], [19], [20], [21], [22], [23]. Most of those methods compute weighted averages using weights based on the exponential of a distance between noisy and clean (selected from the external dataset) patches. A method from this family for Gaussian denoising is the so-called external *non-local-means* (NLM), in which the distance is computed by the ℓ_2 norm of the difference between patches divided by the noise variance [24], [25]. This estimate is shown to converge to the MMSE estimate as the number of samples approach infinity [24]. The name "external NLM" is inspired by the well-known (internal) NLM [6], in which the same weighted averaging of patches is

The authors are with the Instituto de Telecomunicações and Instituto Superior Técnico, University of Lisbon, Portugal.

The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7-PEOPLE-2013-ITN) under grant agreement n 607290 SpaRTaN.

1057-7149 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2912122, IEEE Transactions on Image Processing

computed, but with the patches from the noisy image itself and the distance in the exponential is divided by a hand-tuned parameter. However, external NLM is computationally demanding, as reported in [24], due to the need to use a large external dataset of clean patches[1].

In many sampling methods, a selected group of similar patches from the external datasets is used for restoring a patch [19], [20], [26]. This requires computing the distance between the patches of the image to be restored and those in the external dataset, which, similary to external NLM, in the case of large-scale external datasets, involves a huge computational cost. To address this hurdle, some approaches have focused on accelerating the patch matching computation [27], [28], [29], [21]. Those methods are often heuristic and rely on hierarchical approaches, such as kd-trees, to approximately find the nearest patches based on the ℓ_2 distance. In [23], a sampling approach called Monte-Carlo NLM (MC-NLM) was proposed, which uses approaches form large deviation theory to speed-up external NLM. However, all those methods are hard-wired to deal with Gaussian noise and the generalization to Poisson (or other types of) noise has not been truly addressed.

In the context of image denoising based on external datasets, a recent class of methods has focused on using class-specific datasets [18], [30], [31], [32]. Those methods exploit the fact that, in many cases, the noisy image is known (or can be easily identified) to belong to a certain class, such as face, text, fingerprint, or some type of medical image.

Importance sampling (IS) belongs to the *Monte-Carlo* (MC) family of methods to approximate multi-dimensional expectations/integrals [33], [34]. It is often used in situations where sampling directly from a distribution is difficult, or when reducing the variance of MC estimation is required. The general approach is to sample from a different distribution, called the *proposal distribution*, instead of the *target distribution*, and correcting each function value with a weight that depends on the ratio between the two distributions. If the normalization constant of the proposal or the target distribution (or both) is unknown, a version of IS called *self-normalized IS* (SNIS) can be used, which dispenses with the knowledge of these normalization constants.

In this paper, we propose a new image restoration method based on SNIS, which is applicable to large-scale external datasets and general observation models, including any type of non-Gaussian noise for which the distribution can be computed. The proposed approach approximates the MMSE estimate of the image patches using SNIS applied to a set of samples from the external dataset. Instead of sampling from the posterior distribution, which is unknown, we use, as the proposal distribution, a mixture of densities derived from clustered patches of the external dataset. The mixture distribution is chosen such that it maximizes the similarity to the optimal sampling distribution of SNIS. The method is non-parametric, with the samples being directly obtained from the external dataset, and can be applied with both class-specific and generic image datasets.

The experiments reported show that the proposed method yields state-of-the-art performance under Poissonian and Gaussian noise, in the class-specifi setting, where a dataset of images from the same class is available. The performance of the proposed method is also shown in inpainting problems with different types of noise. In the case of restoration of generic images, the proposed method can be used to accelerate external NLM, thus enabling the use of the proposed method efficiently on large-scale datasets.

The paper is organized as follows. Section 2 reviews the use of IS for approximating multivariate expectations (integrals), for later use in the MMSE estimation of image patches. Section III describes the proposed method. Section 4 reports experimental results on class-specific image restoration and restoration using large scale datasets. Finally, Section 5 ends the paper with a few concluding remarks and pointers to future work.

II. Importance Sampling

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a function, $\mathbf{Z}$ a random vector with pdf p_Z (often termed *target distribution* in the context of IS), and consider the goal of computing the expected value of $f(\mathbf{Z})$, here simply denoted as μ,

$$\mu = \mathbb{E}[f(\mathbf{Z})] = \int_{\mathbf{z}\in\mathbb{R}^n} f(\mathbf{z})\, p_Z(\mathbf{z})\, d\mathbf{z}. \tag{4}$$

It is very often the case that this integral cannot be computed analytically and that, due to the large dimensionality of $\mathbf{Z}$ (large n), using classical numerical integration techniques is also impractical or even infeasible. A possible solution is to approximate the integral in (4) using MC methods. Let $\mathbf{Z}_1, \ldots, \mathbf{Z}_N \sim p_Z$ be a *random sample* of size N following p_Z (*i.e.*, a set of N i.i.d. random variables with pdf p_Z), and $\widehat{\mu}_N^{MC}$ be the sample average random variable

$$\widehat{\mu}_N^{MC} = \widehat{\mathbb{E}}_N^{MC}[f(\mathbf{Z})] = \frac{1}{N}\sum_{j=1}^{N} f(\mathbf{Z}_j). \tag{5}$$

Clearly, this estimator is unbiased, *i.e.*, $\mathbb{E}[\widehat{\mu}_N^{MC}] = \mathbb{E}[f(\mathbf{Z})]$. Moreover, the strong law of large numbers asserts that it is also *consistent*, *i.e.*,

$$\widehat{\mathbb{E}}_N^{MC}[f(\mathbf{Z})] \xrightarrow[N\to\infty]{\text{a.s.}} \mathbb{E}[f(\mathbf{Z})]$$

(where a.s. stands for "almost surely"), provided that $\mathbb{E}|f(\mathbf{Z})| < \infty$ [35, Theorem 5.18].

In cases where sampling from the distribution p_Z is intractable, or the estimator (5) has a large variance for a specific number of samples N, the IS approach stands as a potential alternative [33], [36]. IS generates samples from a tractable distribution q_Z, termed *proposal distribution*, and approximates the expectation (4) by the weighted average

$$\widehat{\mu}_N^{IS} = \widehat{\mathbb{E}}_N^{IS}[f(\mathbf{Z})] = \frac{1}{N}\sum_{j=1}^{N} \frac{p_Z(\mathbf{Z}_j)}{q_Z(\mathbf{Z}_j)} f(\mathbf{Z}_j), \tag{6}$$

where here $\mathbf{Z}_1, \ldots, \mathbf{Z}_N \sim q_Z$, and we are assuming that $q_Z(\mathbf{z}) > 0$, whenever $f(\mathbf{z})\, p_Z(\mathbf{z}) \neq \mathbf{0}$. Similarly to the plain

[1]There are some variants of [24] which are also called external NLM. However, in this paper, by external NLM, we mean the method in [24], which is shown to converge to the MMSE estimate.

1057-7149 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2912122, IEEE Transactions on Image Processing

MC estimator (5), the IS estimator (6) is also unbiased and consistent, under the above-mentioned condition [33], [36].

When only unnormalized versions of p_Z and q_Z are available, a self-normalized version of IS (SNIS) may be used. Let $p_Z = (1/b)\,\tilde{p}_Z$ and $q_Z = (1/c)\,\tilde{q}_Z$, where $\tilde{p}_Z$ and $\tilde{q}_Z$ are unnormalized distributions, and $b > 0$ and $c > 0$ are unknown constants. Assuming that $q_Z(\mathbf{z}) > 0$ whenever $p_Z(\mathbf{z}) > 0$, the expectation in (4) may be approximated by

$$\widehat{\mu}_N^{SNIS} = \widehat{\mathbb{E}}_N^{SNIS}[f(\mathbf{Z})] = \frac{\sum_{j=1}^{N} f(\mathbf{Z}_j) w(\mathbf{Z}_j)}{\sum_{j=1}^{N} w(\mathbf{Z}_j)}, \tag{7}$$

where, as above, $\mathbf{Z}_1, \ldots, \mathbf{Z}_N \sim q_Z$, and

$$w(\mathbf{z}) = \frac{\tilde{p}_Z(\mathbf{z})}{\tilde{q}_Z(\mathbf{z})} \tag{8}$$

are termed the *importance weights* [33], [36], [37]. Notice that if instead of the importance weights given by (8), we used

$$\tilde{w}(\mathbf{z}) = \frac{c\,\tilde{p}_Z(\mathbf{z})}{b\,\tilde{q}_Z(\mathbf{z})} = \frac{p_Z(\mathbf{z})}{q_Z(\mathbf{z})},$$

nothing would change since the factor c/b appears both in the numerator and the denominator of (7), thus being canceled out. The SNIS estimator can be shown to be biased, but consistent, under the above-mentioned conditions [36], [37].

The performance of IS and its SNIS variant depends critically on the proposal distribution q_Z. One may, for example, seek the proposal distribution q_Z^* that, for a given sample size, minimizes the mean square error (MSE); that is,

$$q_Z^* = \arg\min_{q} \mathbb{E}\big[\|\widehat{\mu}_N(q) - \mu\|_2^2\big], \tag{9}$$

where $\widehat{\mu}_N(q)$ refers to the estimators $\widehat{\mu}_N^{IS}$ or $\widehat{\mu}_N^{SNIS}$, using the proposal distribution q. For the IS estimator (6), the solution of (9) is (see proof in [38])

$$q_Z^*(\mathbf{z}) \propto |f(\mathbf{z})|\; p_Z(\mathbf{z}), \tag{10}$$

whereas for SNIS, the optimal sampling density is

$$q_Z^*(\mathbf{z}) \propto |f(\mathbf{z}) - \mu|\; p_Z(\mathbf{z}). \tag{11}$$

These results are obtained via calculus of variations [36, Ch. 2] and used in [37], [39]. The distributions q_Z^* in (10) and (11) are also the minimizers of the asymptotic variance of the corresponding IS and SNIS estimators [39]. In the case of SNIS, obtaining the optimal sampling distribution (11) requires knowing μ, the estimation of which is precisely the goal of SNIS, thus the result cannot be directly applied.

Estimation of the conditional expectations has been addressed using SNIS. Two well-known methods for this purpose are the *population Monte Carlo* (PMC) [40] and *adaptive IS* (AIS) [41]. The proposal distribution in those methods is a mixture of densities designed to guarantee consistency of the SNIS estimator. The parameters of the mixtures are iteratively updated along the iterations, based on the importance weights. However, those methods require sampling from a specific parametric distribution. In several applications, training data is available and can be used as samples. In those cases, the exact distribution of the data is unknown, and fitting any parametric distribution would be an approximation to the true distribution. Consequently, it would be more natural to sample directly from the data, rather than from an approximate fitted distribution. In the next section, we propose a SNIS-based approach that exploits this idea for patch-based image restoration.

III. Proposed Method

A. Introduction

As already mentioned, image restoration based on sampling from a large-scale dataset is computationally complex. In this section, we propose a method for image restoration based on SNIS, which can be efficiently implemented for large-scale and/or class-specific external datasets.

Let $\mathbf{Y} \in \mathbb{R}^n$ denote a random vector associated with the noisy image patches, of size $\sqrt{n} \times \sqrt{n}$, $\mathbf{X} \in \mathbb{R}^n$ the random vector associated with the clean patches corresponding to $\mathbf{Y}$, and $\mathbf{y} \in \mathbb{R}^n$ an observed noise patch (*i.e.*, a sample of $\mathbf{Y}$). Consider the goal of estimating the central pixel of the clean patch, denoted $\mathbf{x}_c$ (leaving aside for now the issue of how the patches are extracted and its estimates combined). As estimation criterion, we adopt the *minimum mean square error* (MMSE), which is well-known to yield the posterior expectation,

$$\widehat{\mathbf{x}}_c = \mathbb{E}[\mathbf{X}_c|\mathbf{Y} = \mathbf{y}] = \int_{\mathbf{x}\in\mathbb{R}^n} \mathbf{x}_c\, p_{X|Y}(\mathbf{x}|\mathbf{y})\, d\mathbf{x}, \tag{12}$$

where $p_{X|Y}(\cdot\,|\mathbf{y})$ denotes the posterior pdf of $\mathbf{X}$ given $\mathbf{Y} = \mathbf{y}$. The expression (12) has the structure of (4), where the function $f(\mathbf{x}) = \mathbf{x}_c$, *i.e.*, $f : \mathbb{R}^n \to \mathbb{R}$ extracts the central pixel of the patch, and the target distribution is the posterior $p_{X|Y}(\cdot\,|\mathbf{y})$.

Computing the integral (12) is, in general, intractable. The best-known exception is the case where $\mathbf{X}$ follows a multivariate Gaussian (or Gaussian mixture) prior distribution and the noise is Gaussian and additive. Assuming that the posterior $p_{X|Y}$ is known, the above integral could be approximated by using plain MC. However, very often, the posterior distribution $p_{X|Y}$ is itself unknown, thus plain MC sampling cannot be applied. Although sampling from the unknown posterior is not feasible, it may be possible to obtain samples from the distribution of clean image patches, available as an external dataset, which may be seen as samples from the prior p_X. This observation motivates the use SNIS to approximate (12).

B. MMSE Estimation by SNIS: Naïve Approach

In order to approximate the MMSE estimate in (12) with a SNIS estimator, recall that the target distribution is the posterior pdf $p_{X|Y}(\cdot\,|\mathbf{y})$, which, according to Bayes law, is given by

$$p_{X|Y}(\mathbf{x}|\mathbf{y}) = p_{Y|X}(\mathbf{y}|\mathbf{x}) p_X(\mathbf{x}) / p_Y(\mathbf{y}), \tag{13}$$

where $p_{Y|X}$ is the conditional pdf of $\mathbf{Y}$ given $\mathbf{X}$, *i.e.*, the likelihood function, p_X is the patch prior, and p_Y is the marginal pdf of $\mathbf{Y}$. In the sequel, since $\mathbf{y}$ is given, we often use the compact notation for the likelihood function

$$l_{\mathbf{y}}(\mathbf{x}) \equiv p_{Y|X}(\mathbf{y}|\mathbf{x}). \tag{14}$$

1057-7149 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2912122, IEEE Transactions on Image Processing

Given N clean patches $\mathbf{x}_1, ..., \mathbf{x}_N$ assumed to be i.i.d. samples from the patch prior p_X, a naïve approach to using SNIS to approximate (12) consists in using p_X as the proposal distribution, leading to

$$\widehat{\widehat{\mathbf{x}}}_c = \frac{\sum_{j=1}^{N} (\mathbf{x}_j)_c \; w(\mathbf{x}_j)}{\sum_{j=1}^{N} w(\mathbf{x}_j)} = \frac{\sum_{j=1}^{N} (\mathbf{x}_j)_c \; l_{\mathbf{y}}(\mathbf{x}_j)}{\sum_{j=1}^{N} l_{\mathbf{y}}(\mathbf{x}_j)}, \tag{15}$$

because the weights are given by

$$\begin{aligned} w(\mathbf{x}_j) &= \frac{p_{X|Y}(\mathbf{x}_j|\mathbf{y})}{p_X(\mathbf{x}_j)} && (16) \\ &\propto \frac{p_{Y|X}(\mathbf{y}|\mathbf{x}_j)\, p_X(\mathbf{x}_j)}{p_X(\mathbf{x}_j)} \\ &= l_{\mathbf{y}}(\mathbf{x}_j), && (17) \end{aligned}$$

where we assume that the prior p_X is non-zero everywhere.

The drawback of this naïve SNIS method is that it needs an extremely large number of external patches to yield decent estimates, since most sampled patches will be very different from the underlying true one, thus the vast majority of weights will be extremely small. In fact, a central issue in any IS method (including SNIS) is finding a proposal distribution that is not too different from the target one, such that the weights are not almost all very small. In the next subsection, we propose an approach to tackle this issue for the patch-based image denoising problem.

Finally, notice that if the noise is additive and Gaussian, $l_{\mathbf{y}}(\mathbf{x}_j) \propto \exp\big(-\|\mathbf{x}_j - \mathbf{y}\|_2^2/(2\,\sigma^2)\big)$, which shows that the patch estimate in standard external NLM methods [24], [25] is nothing more than a SNIS approximation of the MMSE patch estimate. As far as we know, this fact had not been pointed out before in the literature. Moreover, this is obviously suboptimal, since the proposal distribution should be adapted to the target distribution $p_{X|Y}(\cdot\,|\mathbf{y})$; that is, not only the SNIS weights, but also the target distribution, have to depend on the observed $\mathbf{y}$.

C. MMSE Estimation by SNIS: Proposed Approach

As explained in the previous subsection, the target distribution is the posterior pdf $p_{X|Y}(\cdot\,|\mathbf{y})$, with an unnormalized version thereof being simply $l_{\mathbf{y}}\, p_X$. In this subsection, we introduce a proposal distribution that depends on $\mathbf{y}$, which is needed to allow adapting it to the particular target pdf $p_{X|Y}(\cdot\,|\mathbf{y})$ for each $\mathbf{y}$. Following the main proposal distributions in the IS literature [40], [41], we use a mixture distribution. Mixture distributions satisfy the required conditions for the convergence of the sum in (7). The form of mixture adopted in our method, unlike previous proposal distributions, is completely novel in that it does not require a parametric form.

We begin by clustering the external dataset of patches $\{\mathbf{x}_1, ..., \mathbf{x}_L\}$ into K disjoint clusters: $\mathcal{X}_1, ..., \mathcal{X}_K$. Let this clustering induce a partition, $R_1, ..., R_K$ of $\mathbb{R}^n$ that satisfies $\mathcal{X}_k \subset R_k$, for $k = 1, ..., K$. Notice that this partition is obviously not unique, but this will be irrelevant for the proposed method. Using this partition, it is possible to rewrite the prior p_X, of which $\mathbf{x}_1, ..., \mathbf{x}_L$ are assumed to be i.i.d. samples, under the form of a mixture,

$$p_X(\mathbf{x}) = \sum_{k=1}^{K} m_k \; g_k(\mathbf{x}), \tag{18}$$

where $m_k g_k$ is the restriction of p_X to R_k and

$$m_k = \int_{R_k} p_X(\mathbf{x})\, d\mathbf{x} \tag{19}$$

is the corresponding normalization constant, that is,

$$g_k(\mathbf{x}) = \frac{1}{m_k} \begin{cases} p_X(\mathbf{x}) & \text{if } \mathbf{x} \in R_k \\ 0 & \text{if } \mathbf{x} \notin R_k. \end{cases} \tag{20}$$

Naturally, the elements of cluster $\mathcal{X}_k$ are assumed to be samples of a random variable (say $\mathbf{X}_k$) following the pdf g_k.

We suggest the proposal distribution to be a mixture with components $g_1, ..., g_K$ and weights that depend on $\mathbf{y}$, *i.e.*, $\alpha_1(\mathbf{y}), ..., \alpha_K(\mathbf{y})$, to be determined later,

$$\widetilde{q}_X\big(\mathbf{x}; \boldsymbol{\alpha}(\mathbf{y})\big) = \sum_{k=1}^{K} \alpha_k(\mathbf{y}) \; g_k(\mathbf{x}) = p_X(\mathbf{x}) \sum_{k=1}^{K} \frac{\alpha_k(\mathbf{y})}{m_k} \, 1_{R_k}(\mathbf{x}), \tag{21}$$

where $\boldsymbol{\alpha}(\mathbf{y}) = \big(\alpha_1(\mathbf{y}), \ldots, \alpha_K(\mathbf{y})\big)$ are the mixture coefficients of the proposal distribution (which are non-negative and add to one, *i.e.*, $\boldsymbol{\alpha}(\mathbf{y})$ belong to the $(K-1)$-dimensional probability simplex $\Delta^{(K-1)}$), and 1_A denotes the indicator function of some set A, that is, $1_A(\mathbf{x}) = 1$, if $\mathbf{x} \in A$, and 0 otherwise. The notation $\boldsymbol{\alpha}(\mathbf{y})$ is used to stress that these weights will be adapted as a function of $\mathbf{y}$. The resulting SNIS weights for some sample $\mathbf{x}$ are given by

$$w(\mathbf{x}) = \frac{l_{\mathbf{y}}(\mathbf{x})\, p_X(\mathbf{x})}{\widetilde{q}_X(\mathbf{x}; \boldsymbol{\alpha}(\mathbf{y}))} = \frac{l_{\mathbf{y}}(\mathbf{x})}{\sum_{k=1}^{K} \frac{\alpha_k(\mathbf{y})}{m_k} 1_{R_k}(\mathbf{x})}. \tag{22}$$

The choice of optimal weights $\boldsymbol{\alpha}(\mathbf{y})$ will be discussed later in the following subsection, as well as how to sample from $\widetilde{q}_X(\mathbf{x}; \boldsymbol{\alpha}(\mathbf{y}))$ without knowing p_X.

This proposed SNIS estimator has a few distinctive features, namely: **(i)** knowledge of the marginals $p_X(\mathbf{x})$ and $p_Y(\mathbf{y}_i)$ is not needed; **(ii)** any likelihood function $l_{\mathbf{y}}(\mathbf{x})$ can be used; **(iii)** the samples $\mathbf{x}_j \sim \widetilde{q}_X(\cdot; \boldsymbol{\alpha}(\mathbf{y}))$ can be easily obtained from the external dataset as explained below.

In earlier work [42], [43], we proposed IS-based methods for denoising. However, those methods are quite different from the one herein proposed: in those methods, a distribution for each patch is selected from a set of learned distributions, and then external NLM is used.

D. Optimizing the Proposal Distribution

We now address the setting of $\boldsymbol{\alpha}(\mathbf{y})$ in the proposal distribution (21). Our approach is based on (11), which provides the optimal proposal distribution, and on a similarity measure between two probability distributions. Since the target distribution is the posterior $p_{X|Y}(\cdot|\mathbf{y})$, the optimal sampling distribution is

$$q_X^*(\mathbf{x}) \propto |\mathbf{x}_c - \widehat{\mathbf{x}}_c| \; p_{X|Y}(\mathbf{x}\,|\mathbf{y}). \tag{23}$$

1057-7149 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2912122, IEEE Transactions on Image Processing

The optimal proposal distribution q_X^* depends on $\widehat{\mathbf{x}}_c$, which, obviously, we do not have, as it is precisely the goal of the estimation procedure. Later in this section, we will discuss an alternating minimization approach to address this issue. For now, assume that an estimate $\widehat{\mathbf{x}}_c$ is available.

A natural criterion to adjust the weight vector $\boldsymbol{\alpha}(\mathbf{y})$ is to minimize some *distance* measure between $\widetilde{q}_X(\cdot\,;\boldsymbol{\alpha})$ and q^*. An obvious choice would be the Kullback-Leibler divergence [44]; however, it is not symmetric and its computation is not straightforward for some distributions. Here, we use the squared Hellinger distance [45], which, given two probability density functions q and p, is defined as

$$H^2(p,q) = 1 - \int \sqrt{p(\mathbf{x})q(\mathbf{x})}\; d\mathbf{x}. \tag{24}$$

The Hellinger distance is a metric satisfying all the corresponding properties (namely, symmetry and triangle inequality). The application of the Hellinger distance to our setup yields

$$\begin{aligned}\widehat{\boldsymbol{\alpha}} &= \underset{\boldsymbol{\alpha}\in\Delta^{K-1}}{\arg\min}\, H^2(q_X^*, \widetilde{q}_X(\cdot\,;\boldsymbol{\alpha})) \\ &= \underset{\boldsymbol{\alpha}\in\Delta^{K-1}}{\arg\max} \int \sqrt{\widetilde{q}_X(\mathbf{x};\boldsymbol{\alpha})q^*(\mathbf{x})}\; d\mathbf{x}.\end{aligned} \tag{25}$$

It is worth mentioning that the Bhattacharyya distance (another distance measure for probability density functions) leads to the same optimization problem (25), as it is defined as $B(p,q) = -\ln \int \sqrt{p(\mathbf{x})q(\mathbf{x})}\; d\mathbf{x}$ [46].

By inserting (21) and (23) into (25), we obtain

$$\widehat{\boldsymbol{\alpha}} = \underset{\boldsymbol{\alpha}\in\Delta^{K-1}}{\arg\max} \int_{\mathbb{R}^n} \left(\left|\mathbf{x}_c - \widehat{\mathbf{x}}_c\right| l_{\mathbf{y}}(\mathbf{x}) \sum_{k=1}^{K} \frac{\alpha_k}{m_k} 1_{\mathcal{X}_k}(\mathbf{x}) \right)^{\frac{1}{2}} p_X(\mathbf{x}) d\mathbf{x}. \tag{26}$$

To approximate the integral in (26), we resort to MC sampling from p_X, by using M samples, $\mathbf{x}_1, ..., \mathbf{x}_M$, from the available external dataset, yielding

$$\widehat{\boldsymbol{\alpha}} = \underset{\boldsymbol{\alpha}\in\Delta^{K-1}}{\arg\max} \sum_{s=1}^{M} \left(\left|(\mathbf{x}_s)_c - \widehat{\mathbf{x}}_c\right| l_{\mathbf{y}}(\mathbf{x}_s) \sum_{k=1}^{K} \frac{\alpha_k}{m_k} 1_{\mathcal{X}_k}(\mathbf{x}_s) \right)^{1/2}. \tag{27}$$

Notice that the samples $\mathbf{x}_1, ..., \mathbf{x}_M$ can be partitioned according to which cluster $\mathcal{X}_1, ..., \mathcal{X}_K$ each one belongs to. Furthermore, noticing that if $\mathbf{x}_s \in \mathcal{X}_j$, then $1_{\mathcal{X}_j}(\mathbf{x}_s) = 1$, whereas $1_{\mathcal{X}_r}(\mathbf{x}_s) = 0$, for $r \neq j$, allows re-writing (27) as

$$\widehat{\boldsymbol{\alpha}} = \underset{\boldsymbol{\alpha}\in\Delta^{(K-1)}}{\arg\max} \sum_{k=1}^{K} \left(\frac{\alpha_k}{m_k}\right)^{1/2} \sum_{s:\mathbf{x}_s\in\mathcal{X}_k} \left(\left|(\mathbf{x}_s)_c - \widehat{\mathbf{x}}_c\right| l_{\mathbf{y}}(\mathbf{x}_s) \right)^{1/2}. \tag{28}$$

Before proceeding, recall that m_k is given by (19), thus its MC-based estimate is simply $\widehat{m}_k = |\mathcal{X}_k|/L$. Plugging this estimate in the previous expression yields

$$\widehat{\boldsymbol{\alpha}} = \underset{\boldsymbol{\alpha}\in\Delta^{(K-1)}}{\arg\max} \sum_{k=1}^{K} \sqrt{\alpha_k}\; b_k, \tag{29}$$

where

$$b_k = \left(\frac{1}{|\mathcal{X}_k|}\right)^{1/2} \sum_{s:\mathbf{x}_s\in\mathcal{X}_k} \left(\left|(\mathbf{x}_s)_c - \widehat{\mathbf{x}}_c\right| l_{\mathbf{y}}(\mathbf{x}_s) \right)^{1/2}. \tag{30}$$

Finally, the optimal solution of (29) is

$$\widehat{\alpha}_k = \frac{b_k^2}{\sum_{k=1}^{K} b_k^2}, \qquad k = 1, \ldots, K. \tag{31}$$

To see why this is so, consider the following change of variables: $\beta_k = \sqrt{\alpha_k}$, thus $\alpha_k = \beta_k^2$. With this change of variable, problem (29) becomes

$$\widehat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}\in S^{(K-1)}}{\arg\max} \sum_{k=1}^{K} \beta_k\; b_k = \underset{\boldsymbol{\beta}\in S^{(K-1)}}{\arg\max}\; \boldsymbol{\beta}^T \mathbf{b}, \tag{32}$$

where S^{K-1} denotes the unit-radius sphere in $\mathbb{R}^n$. The solution of this problem is well known to be the normalization (to unit norm) of $\mathbf{b}$, that is, $\widehat{\boldsymbol{\beta}} = \mathbf{b}/\|\mathbf{b}\|_2$. Finally, inverting the change of variable yields the solution (31).

Finally, because the optimal $\widehat{\boldsymbol{\alpha}}$ does depend on $\mathbf{y}$ (because $\mathbf{b}$ depends on $\mathbf{y}$), we will recover the notation used in the previous subsection and refer to it as $\widehat{\boldsymbol{\alpha}}(\mathbf{y})$.

E. Sampling and Weighting

After the optimal $\widehat{\boldsymbol{\alpha}}(\mathbf{y})$ has been obtained as described in the previous subsection, we now explain how to obtain N samples from the optimized proposal distribution

$$\widetilde{q}_X\big(\mathbf{x}; \widehat{\boldsymbol{\alpha}}(\mathbf{y})\big) = \sum_{k=1}^{K} \widehat{\alpha}_k(\mathbf{y})\; g_k(\mathbf{x}). \tag{33}$$

Recall that a sample from this finite mixture can be obtained by first sampling from a categorical variable with probabilities $\widehat{\alpha}_1(\mathbf{y}), ..., \widehat{\alpha}_K(\mathbf{y})$, and then sampling from the selected component. If the number of samples N is large, this is approximately equivalent to obtaining $N_k = \text{round}(\widehat{\alpha}_k(\mathbf{y})\; N)$ samples from each component g_k, for $k = 1, ..., K$. The samples from the k-component are simply obtained by randomly sampling from the cluster of clean patches $\mathcal{X}_k$.

Computing the weight of each sample according to (22) is very simple, since only one of the terms in the sum in the denominator is non-zero: the one corresponding the cluster from which that sample was obtained. That is,

$$\mathbf{x} \in \mathcal{X}_k \Rightarrow w(\mathbf{x}) = \frac{m_k\; l_{\mathbf{y}}(\mathbf{x})}{\widehat{\alpha}_k(\mathbf{y})}. \tag{34}$$

F. Dealing with the Unknown $\widehat{\mathbf{x}}_c$

As mentioned before, and is obvious in (26)–(28), obtaining the optimal $\widehat{\boldsymbol{\alpha}}(\mathbf{y})$ requires knowing $(\widehat{\mathbf{x}})_c$, which is precisely the object of the estimation problem. To tackle this issue, we use an iterative approach that alternates between the following two steps, after initializing the estimate $\widehat{\mathbf{x}}_c$ with the noisy observation: **(i)** $\widehat{\boldsymbol{\alpha}}(\mathbf{y})$ is computed via (31); **(ii)** the estimate $\widehat{\mathbf{x}}_c$ is updated via SNIS. These two steps repeated until some convergence criterion is satisfied.

G. Full Patch Estimation

We now consider a variant of the above-proposed method that restores the whole patch, instead of just the central pixel thereof. This variant is faster and yields improved performance

1057-7149 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2912122, IEEE Transactions on Image Processing

for the same running time. Results with the two implementations are provided in Section IV.

Inspired by the patch-based approaches [4], [5], the noisy image is first divided into overlapping patches[2], then, instead of restoring merely the central pixel, the whole patch is denoised; finally, the patches are returned to the original positions and are averaged in overlapping pixels. In the restoration step, we extend the central pixel estimate (15) to the whole patch. Notice that the estimate in (15) is still valid if $(\mathbf{x}_j)_c$ is replaced by $(\mathbf{x}_j)_d$, where d denotes any pixel index in a patch, not necessarily the central one. This modification raises an issue: the optimal sampling distribution in (11) is only valid if the range of function f is $\mathbb{R}$, thus it is not directly applicable if the central pixel is replaced by the whole patch. We address this issue by choosing $\boldsymbol{\alpha}$ to maximizes the similarity measures for all pixels in a patch, on average, *i.e.*,

$$\widehat{\boldsymbol{\alpha}} = \arg\max_{\boldsymbol{\alpha}\in\Delta^{K-1}} \sum_{d=1}^{n} \int_{\mathbb{R}^n} \left(\left|(\mathbf{x})_d - \widehat{\mathbf{x}}_d\right| l_{\mathbf{y}}(\mathbf{x}) \sum_{k=1}^{K} \frac{\alpha_k}{m_k} 1_{\mathcal{X}_k}(\mathbf{x}) \right)^{\frac{1}{2}} p_X(\mathbf{x}) d\mathbf{x}.$$

Following the same rational discussed in the previous section, and a simple rearrangement of summations, in this case $\boldsymbol{\alpha}$ is given by (31), where

$$b_k = \left(\frac{1}{|\mathcal{X}_k|}\right)^{1/2} \sum_{d=1}^{n} \sum_{s:\mathbf{x}_s\in\mathcal{X}_k} \left(\left|(\mathbf{x}_s)_d - \widehat{\mathbf{x}}_d\right| l_{\mathbf{y}}(\mathbf{x}_s) \right)^{\frac{1}{2}}.$$

The motivation for the whole-patch approach is that the computational bottleneck of the proposed method (as in NLM), is the computation of the likelihood function. Using the whole-patch procedure, computing the weights for the whole patch requires the same number of computations of the likelihood as for just the central pixel. Furthermore, by adjusting the stride (displacement between consecutive patches), it is possible to control the trade-off between computational cost and performance. In Section IV, we report result with both the whole-patch and central-pixel method. The algorithm with the central-pixel-based method is shown in Fig. 1, while Figure 2 summarizes the proposed whole-patch scheme.

- For each pixel i in the image
  - Extract the patch $\mathbf{y}_i$ with the pixel in center, and iterate for L times:
    - Compute the mixture coefficients $\boldsymbol{\alpha}(\mathbf{y}_i)$ by (31).
    - Extract the patches $\mathbf{x}_j$'s from the mixture distribution $\widetilde{q}_X\big(\mathbf{x}; \boldsymbol{\alpha}(\mathbf{y}_i)\big)$.
    - Estimate the central pixel by the weighted average (15).

Fig. 1. The algorithm of the proposed SNIS method for central-pixel estimation.

- Divide noisy image into overlapping patches $\mathbf{y}_i$, $i = 1, \dots I$.
- For each patch $\mathbf{y}_i$, iterate for L times:
  - Compute the mixture coefficients $\boldsymbol{\alpha}$ by (31).
  - Extract the patches $\mathbf{x}_j$'s from the mixture distribution $\widetilde{q}_X\big(\mathbf{x}; \boldsymbol{\alpha}(\mathbf{y}_i)\big)$.
  - Compute the weighted average of patches using (15) for all pixels of the patch.
- Return the restored patches to the original position in the image and average in the overlapping pixels.

Fig. 2. The algorithm of the proposed SNIS method for whole-patch estimation.

H. Practical implementation

For clustering the external dataset, any (hard clustering) algorithm, such as k-means, can be used, since the proposed restoration algorithm does not depend critically on the clustering. In our implementation, we use the *classification-EM* (CEM) algorithm [47], which fits K multivariate Gaussian distributions to the data and considers the samples assigned to each of these distribution as a cluster.

Based on the above considerations, it is clear that there is no need for a parametric form of the mixture distribution $\widetilde{q}(\cdot\,; \boldsymbol{\alpha})$ or even the distribution of natural images p_X. The proposed approach may be seen as new general method based on IS, which, unlike other methods such as [41], [48], does not require any parametric proposal distribution, and can potentially be used with other sources of data. Another important feature of the proposed method is that it requires merely the evaluation of the likelihood (or an unnormalized version thereof), which is usually (assumed) known and easy to compute [49].

Implementation of the proposed method is computationally expensive if all the samples in the external dataset are used for obtaining the coefficients $\boldsymbol{\alpha}$ and estimating each patch. However, as shown in the result section, a very limited number of samples M (less than 1% of the whole dataset) suffices to obtain a good estimate of $\boldsymbol{\alpha}$. Our approach then uses a limited number N of so-called *important* samples, derived from the proposal distribution to estimate each patch. Consequently, the total number of samples (*i.e.* $N + M$) is much smaller than what is typically used in external NLM. We defer further discussion on the computational complexity to Section IV. We will also show that the proposed SNIS approach performs better than similar efficient approaches for large-scale Gaussian denoising, such as MC-NLM [23].

IV. Experimental results

In this section, we assess the performance of the proposed method on both class-specific and large-scale generic image datasets.

A. Class-specific image restoration

In this subsection, we apply the proposed method to several problems where a dataset of the same class of the noisy

[2]In addition, we will test different strides (i.e., shifts between consecutive extracted patches), which control the trade-off between time complexity and denoising performance.

1057-7149 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2912122, IEEE Transactions on Image Processing

image is available; in fact, in many applications, the image class is known or can be obtained (*e.g.*, an image of a face, a fingerprint, text, etc.). This scenario has recently received considerable attention [18], [30], [31], [32].

The first set of experiments addresses denoising, where the observed image is contaminated by either Gaussian or Poissonian noise. The second set of experiments addresses image inpainting, *i.e.*, some pixels in the image are missing. The combination of noise with missing pixels is also considered. The Gore face dataset [50] and the MNIST handwritten digits dataset [51] are adopted as external datasets. For the text dataset, we extracted images from different parts of typed text documents. For the face and text dataset, 5 images are considered as the test image and the others are used as external. For the MNIST handwritten dataset, we used the splitting into training and testing that is provided in the corresponding website. Regarding the proposed method, we have found that, to achieve the best performance, different number of clusters are needed for different datasets. For example, for face, text, and handwritten digit datasets, we use 20, 30, and 50 clusters of 9×9 patches, respectively.

All the results reported in this section were obtained using the whole-patch approach described in Figure 2, with the following parameter settings. A small number of random samples for each of the two stages (*i.e.*, N and M) is used. Specifically, we set $M = 900$, which is less than 1% of the samples in each external datasets and $N = 300$. Therefore, for each degraded patch, a total of 1200 sample patches are used, which involves a computational complexity roughly similar to an internal non-local denoising procedure, with the patches constrained approximately to a 35×35 search window. The number of iterations L in the alternating minimization approach was set to 3. The stride for selecting patches in the noisy image is set to 2, to further reduce the computational cost. We empirically found that this value is sufficient to achieve competitive denoising results, while keeping the computational cost reasonably low. More details about the computational cost of the proposed algorithm will be discussed in the subsection related to the large-scale generic dataset.

1) Gaussian denoising: The first set of experiments addresses image denoising under i.i.d. Gaussian additive noise, *i.e.*, the observation model is given by (1) with $\mathbf{H} = \mathbf{I}$, thus the likelihood has the form in (2). Table I shows denoising *peak-signal-to-noise-ratio* (PSNR) results (in dB) for various values of the noise standard deviation σ and for the MNIST and text datasets. In addition to the results for the proposed SNIS method, the table also shows results for the methods EPLL (generic) and BM3D (generic) and [30] (class-specific). The proposed method outperforms, not only the methods designed for generic images, but also the method proposed in [30], which is class-specific. An example of a denoising result for the face dataset is shown in Figure 3.

2) Poisson Denoising: In these experiments, we generate images corrupted by Poissonian noise, according to model (3) with $\mathbf{H} = \mathbf{I}$. Table II shows denoising PSNR results averaged over 5 test images for the above-mentioned face and text datasets. The performance of SNIS is compared with those of NL-PCA [1], VST+BM3D [2], and P4IP [52], in low SNR regimes. All those methods are generic, as we know of no other class-specific Poisson denoising method performing well on our dataset[3]. The results show that SNIS noticeably outperforms other methods. An example of denoising of 5 digits from the MNIST dataset is shown in Figure 4.

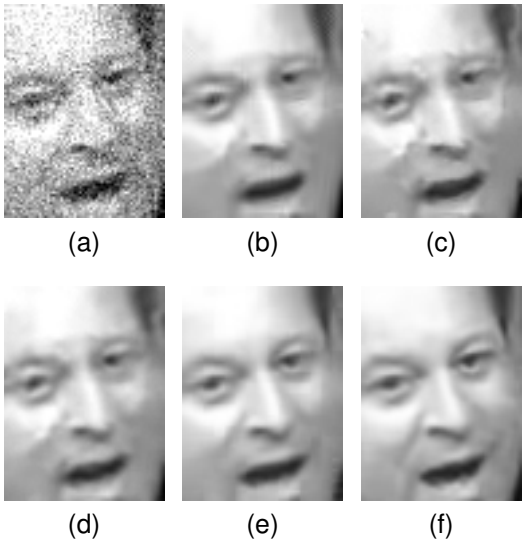

Fig. 3. Example of denoising for a face image in the Gore dataset ($\sigma = 30$): (a) noisy image; (b) BM3D (PSNR=29.80 dB); (c) EPLL (PSNR=28.28 dB); (d) Class specific EPLL (PSNR=29.98 dB); (e) Class-specific denoising in [30] (PSNR=32.98 dB); (f) this work (PSNR=34.11 dB).

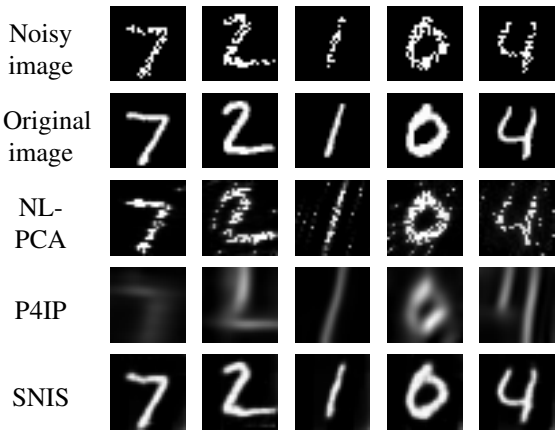

Fig. 4. Examples of denoising of digits from the MNIST dataset contaminated with Poisson noise. The maximum pixel value of the original image is 2. PSNR values in dB for NL-PCA: 13.55, 11.44,14,47,10.62, 13.43/ P4IP: 12.86, 12.37, 17.38, 12.93, 13.92/ SNIS: 21.64, 19.87, 23.82, 19.97, 19.84.

3) Image Inpainting in Additive Gaussian Noise: Image inpainting is the problem of recovering images in which some pixels are missing. In this case, $\mathbf{H}$ is a diagonal matrix in which the diagonal entries are either zero or one, corresponding to the missing or available pixels, respectively. In the noiseless case, the value of σ in (2) can be taken very small. However, for a very small value of σ, the likelihood function can become numerically zero for some observed patches; it may even happen that for some noisy patches, all the sampled patches have (numerically) zero likelihood. In this case, the

[3]There is one method based on deep networks [53], which is learned on much larger datasets, such as ImageNet.

1057-7149 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2912122, IEEE Transactions on Image Processing

TABLE I
GAUSSIAN DENOISING: PSNR (DB) AVERAGED OVER 5 TEST IMAGES. DATASETS: MNIST [50] AND TEXT.

	$\sigma = 20$		$\sigma = 30$		$\sigma = 40$		$\sigma = 50$	
	MNIST	text	MNIST	text	MNIST	text	MNIST	text
BM3D	28.30	28.13	25.39	24.95	21.81	22.55	26.98	20.91
EPLL (generic)	28.51	28.15	25.64	25.21	22.07	23.15	22.77	21.72
Luo et. al. [30]	27.03	27.52	26.34	27.44	24.81	26.29	24.79	25.02
SNIS	**29.46**	**29.98**	**28.62**	**29.19**	**26.01**	**28.31**	**27.75**	**27.29**

TABLE II
CLASS-SPECIFIC POISSON DENOISING FOR DIFFERENT PEAK VALUES SHOWN IN THE FIRST ROW. THE REPORTED PSNR (DB) IS AVERAGED OVER 5 TEST IMAGES.

	10		5		2		1	
	Face	text	Face	text	Face	text	Face	text
NL-PCA	25.01	22.16	23.80	19.66	22.87	14.92	19.69	12.70
VST+BM3D	25.41	23.15	24.79	20.96	23.70	16.29	20.80	13.89
P4IP	25.84	23.51	24.88	21.19	23.78	17.22	20.03	14.12
SNIS	**27.40**	**24.32**	**25.78**	**23.83**	**23.95**	**22.90**	**21.31**	**21.02**

TABLE III
IMAGE INPAINTING RESULTS FOR THE GORE FACE AND MNIST DATASETS, DIFFERENT PERCENTAGE OF RANDOMLY AVAILABLE PIXELS. THE REPORTED PSNR IS AVERAGED OVER FIVE TEST IMAGES.

	50%		20%		10%	
	face	MNIST	face	MNIST	face	MNIST
KR	32.79	18.26	30.78	16.58	24.51	13.82
FOE	41.72	26.02	31.21	18.58	26.18	15.60
EPLL	39.36	24.50	30.25	17.98	25.20	15.15
SNIS	**48.30**	**26.33**	**37.47**	**22.22**	**29.26**	**17.20**

estimate in (2) is not defined. In order to avoid this numerical problem, for each observed patch, the value of σ is initialized to a very small value, and for the patches in which the obtained weights are all zero, σ iteratively increased to achieve at least one non-zero weight. The results of image inpainting for different percentages of randomly missing pixels are reported in Table III. The proposed method is compared to EPLL [11], kernel regression (KR) [54], and the field of experts (FoE) [55] methods, all of which are generic. An example of a restored text image obtained by the proposed method is compared to other methods in Figure 5.

Finally, we evaluate the proposed method in a more general observation models, which combines noise with loss of pixels. An example of the restored images is shown in Fig. 6 for the Gaussian and Poisson noise models. For the Poisson case, some weights are ambiguous as the term $((\mathbf{Hx})_l)^{(\mathbf{y})_l}$ in (3) becomes zero in both base and exponent. In this case, we set these entries to 1 as no information is available for the missing pixel. Any other constant value is also possible, since it cancels out in the weighted average.

B. Generic image denoising

If the class of the image is known, the prior p_X is well-adapted to the noisy image, resulting in state-of-the-art performance. However, for generic images, the prior from natural image patches is not specifically adapted to the image. As discussed in [24], the performance of BM3D denoising is close to the MMSE estimate for generic images. However, here we focus on another major challenge of using sampling methods in the generic datasets: the very large number of samples in the external datasets. Unlike in the class-specific case, where the number of samples in the external datasets is limited, generic images require a very large dataset to achieve good denoising performance. In this case, sampling methods such as external NLM become computationally very expensive [24].

Although naive SNIS and the proposed method both use the likelihood l_y in the restoration step, the main difference is in the sampling step in which the samples in our method are chosen based on the noisy patch $\mathbf{y}$ rather than being independent of it. It is well-known that the computational cost of the NLM algorithm for denoising one pixel is $O(Sn)$, where S is the number of patch samples and n is the number of pixels in a patch [56]. So, the computational complexity can roughly be measured by the number of processed patches from the external dataset. As a result, several previous methods limit the number of processed samples from the dataset to speed-up denoising. Our method belongs to this group and, in this section, we compare it to other methods in this class. One approach would be simply selecting randomly a specific number of samples S from the dataset and then compute the weighted average (15) to estimate the pixel. We call this method *uniform sampling* (following [56]). Another approach is MC-NLM [56], where a Bernoulli sampling probability is assigned to each patch based on large-deviation theory techniques. The patches are then selected based on this sampling distribution and the weighted average is computed. In our method, the number of processed patches is computed as the sum of the number of patches processed for the mixture coefficients M and the number of patches used for estimation, N, *i.e.*, $S = N + M$. For the MC-NLM, apart from the number of samples, the complexity of obtaining the sampling distribution for each patch should also be considered.

1057-7149 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2912122, IEEE Transactions on Image Processing

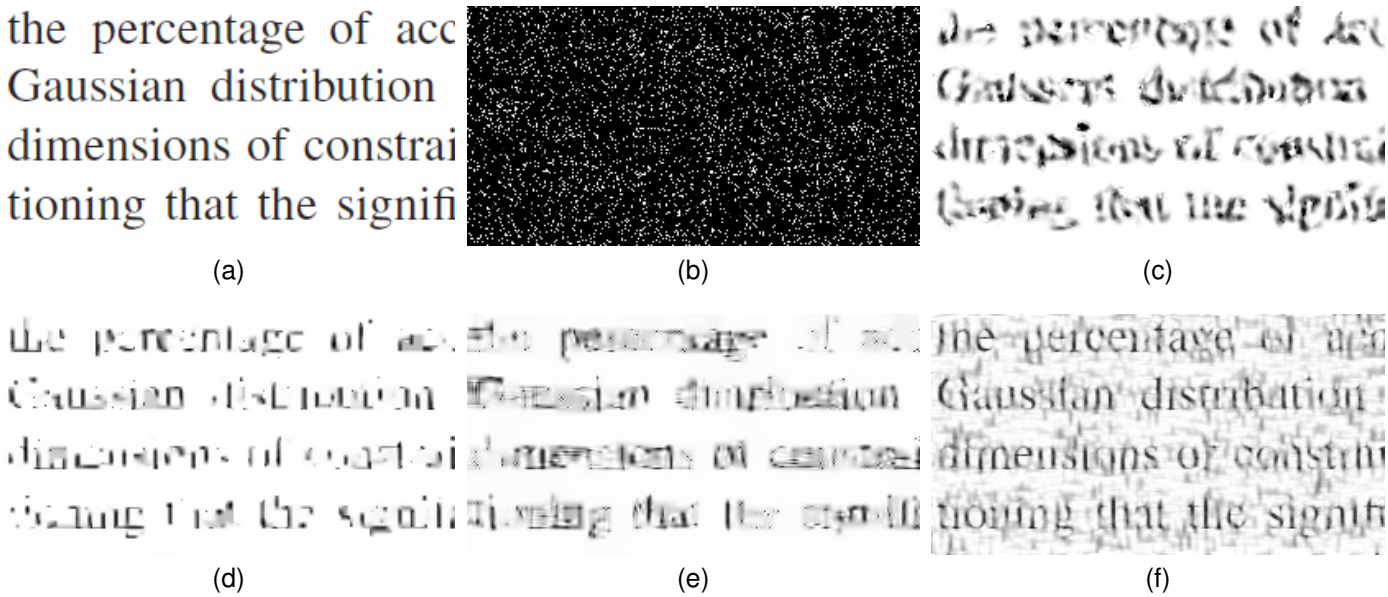

Fig. 5. Image inpainting results: (a) original image; (b) degraded image (10% of pixels are available); (c) kernel regression (PSNR=13.91 dB); (d) FOE (PSNR=15.24 dB); (e) EPLL (PSNR=14.85 dB); (f) SNIS (PSNR=16.63 dB).

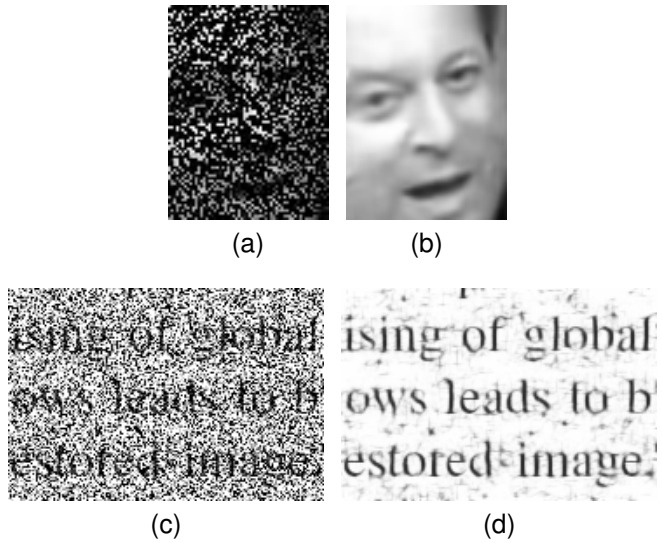

Fig. 6. Examples of recovering images with missing pixels and contaminated by noise using SNIS. (a) Degraded image: 70% randomly available pixels with additive i.i.d. Gaussian noise with standard deviation 15. (b) Restored image (PSNR=33.26 dB). (c) Degraded image with peak of 5 contaminated by Poisson noise and observation of 80% of pixels observed (PSNR=4.92 dB). (d) Restored image (PSNR=20.18 dB)

The experiments in this section use 2 million (2×10^6) external clean patches extracted from the generic image dataset in [57]. Figures 7 and 8 show two denoising examples for Gaussian and Poisson noise, respectively. In both cases, our method is compared to the uniform sampling and the exact MMSE [24], with the following parameters. For SNIS, the patches in the external dataset are divided into 220 clusters; we use two iterations with $M = 2200$ and $N = 300$. The number of samples for uniform sampling was set to 5000 (the same total number of samples as our method). The exact MMSE estimate is equivalent to external NLM in the Gaussian noise case. It can be seen that, in both experiments, the proposed method outperforms the uniform sampling by a noticeable margin, while the CPU time is roughly the same. This is mainly due to the sampling procedure proposed in this paper, which is based on the optimal sampling distribution for SNIS. The exact MMSE denoising, however, outperforms both methods but with a huge increase in computational cost. It can also be seen that the performance of the proposed method with a limited number of samples is not far from the optimal MMSE.

In a final experiment, the discussed methods are used to estimate central pixels of 2000 patches used in [56], under Gaussian noise with $\sigma = 18$. As discussed above, for the methods which use a subset of patch samples in the weighted average, the computational cost is determined by the number of samples used. Figure 9 plots PSNR as a function of the number of sampled patches, up to 6×10^4. This range is suitable for computationally efficient denoising. Each point in the plot indicates one-stage evaluation of the methods consisting of processing 3000 patches. For the proposed method, the central pixel estimation algorithm in 1 is used, and the horizontal axis indicates the total number of patches processed, *i.e.*, for obtaining the mixture plus the patches used for estimation. In each stage, 0.6 of the total number of patches are used for updating the mixture coefficients $\boldsymbol{\alpha}$, and the others are used for updating the estimate. At each stage, the estimated pixel is replaced with $\hat{\mathbf{x}}_c$ in (28) in order to obtain $\boldsymbol{\alpha}$. It can be seen that SNIS outperforms uniform sampling and MC-NLM [56] for all the numbers of patches considered.

V. Conclusion and Future Works

We introduced a new *self-normalized importance sampling* (SNIS) approach for image denoising, using samples from an external dataset of clean patches. We showed that the *external non-local means* (ENLM) algorithm is a special case of the proposed method for Gaussian noise. Our method has the

1057-7149 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2912122, IEEE Transactions on Image Processing

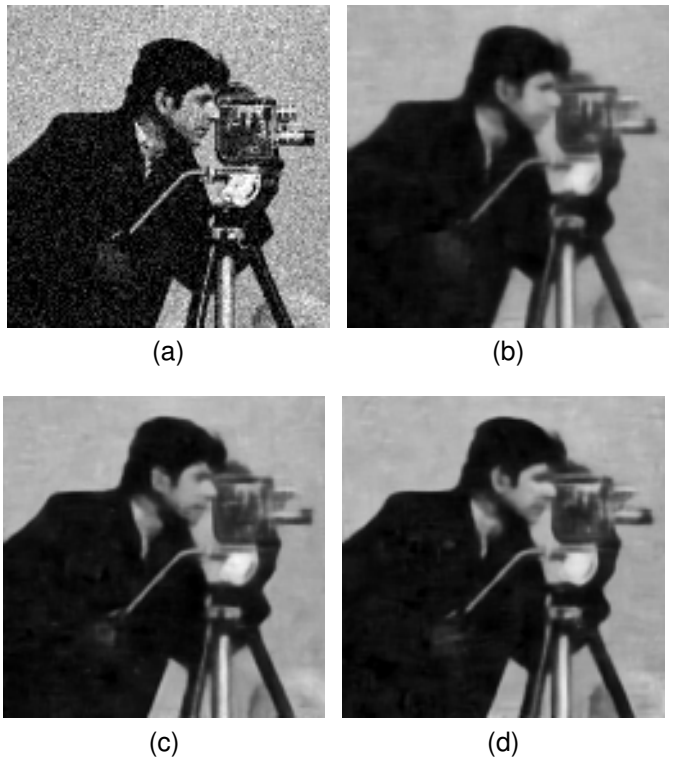

Fig. 7. Denoising example (Gaussian noise): (a) noisy image ($\sigma = 30$); (b) uniform sampling: PSNR=23.08 dB, CPU time = 92 seconds; (c) SNIS: PSNR=24.16 dB, CPU time=94 seconds; (d) exact MMSE: PSNR=24.78 dB, CPU time=3 hours.

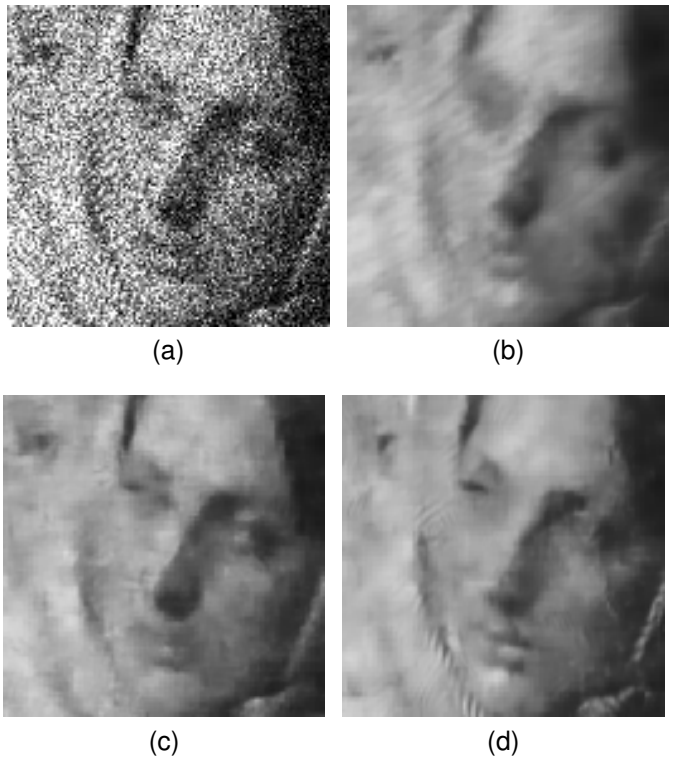

Fig. 8. Denoising a part of the Barbara image with Poisson noise: (a) noisy image (peak=10); (b) uniform sampling, PSNR=24.63 dB, CPU time = 157 seconds; (c) SNIS, PSNR=25.16 dB, CPU time=159 seconds; (d) exact MMSE, PSNR=25.35 dB, CPU time = 3.5 hours.

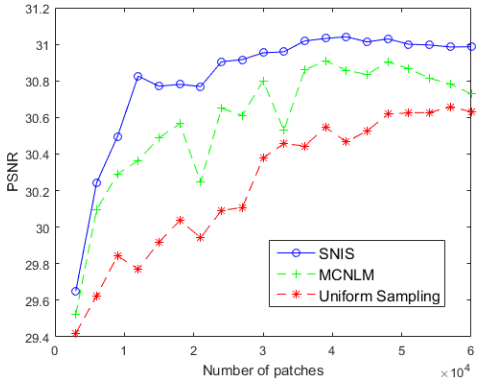

Fig. 9. Gaussian denoising of the central pixels of 2000 noisy image patches ($\sigma = 20$) using a large-scale dataset. PSNR obtained by different methods as a function of the number of processed patches from the dataset.

following main advantages: a) it applies seamlessly to any data observation model for which the likelihood function can be computed, namely Gaussian and Poisson noise; b) it is applicable to large-scale external datasets; and c) it yields state-of-the-art results for the tested class-specific datasets.

How to adaptively determine the numbers of patches (M and N) used in the two stages of the algorithm is an open question to which we will devote future work. We also plan to extend the proposed optimized SNIS-based approach to address other inverse problems, such as image phase estimation from noisy observations.

REFERENCES

[1] J. Salmon, Z. Harmany, C. Deledalle, and R. Willett, "Poisson noise reduction with non-local PCA," *Journal of Mathematical Imaging and Vision*, vol. 48, no. 2, pp. 279–294, 2014.

[2] M. Makitalo and A. Foi, "Optimal inversion of the Anscombe transformation in low-count Poisson image denoising," *IEEE Trans. on Image Processing*, vol. 20, no. 1, pp. 99–109, 2011.

[3] M. Niknejad and M. A. T. Figueiredo, "Poisson image denoising using best linear prediction: A post-processing framework," *arXiv preprint arXiv:1803.00389*, 2018.

[4] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Trans. on image processing*, vol. 16, no. 8, pp. 2080–2095, 2007.

[5] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. on Image processing*, vol. 15, no. 12, pp. 3736–3745, 2006.

[6] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 2005, pp. 60–65.

[7] W. Dong, G. Shi, and X. Li, "Nonlocal image restoration with bilateral variance estimation: a low-rank approach," *IEEE Trans. on image processing*, vol. 22, no. 2, pp. 700–711, 2013.

[8] M. Niknejad, H. Rabbani, and M. Babaie-Zadeh, "Image restoration using Gaussian mixture models with spatially constrained patch clustering," *IEEE Trans. on Image Processing*, vol. 24, pp. 3624–3636, 2015.

[9] A. Teodoro, M. Almeida, and M. Figueiredo, "Single-frame image denoising and inpainting using Gaussian mixtures," in *4th International Conf. on Pattern Recognition Applications and Methods*, 2015.

[10] E. Luo, S. H. Chan, and T. Q. Nguyen, "Image denoising by targeted external databases," in *2014 IEEE International Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 2450–2454.

[11] D. Zoran and Y. Weiss, "From learning models of natural image patches to whole image restoration," in *2011 International Conf. on Computer Vision (ICCV)*, 2011, pp. 479–486.

[12] F. Chen, L. Zhang, and H. Yu, "External patch prior guided internal clustering for image denoising," in *Proceedings of the IEEE International Conf. on Computer Vision*, 2015, pp. 603–611.

1057-7149 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2912122, IEEE Transactions on Image Processing

[13] I. Mosseri, M. Zontak, and M. Irani, "Combining the power of internal and external denoising," in *IEEE International Conf. on Computational Photography*, 2013, pp. 1–9.

[14] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. on Image Processing*, 2017.

[15] Y. Chen and T. Pock, "Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1256–1272, 2017.

[16] J. Jiao, W. Tu, S. He, and R. W. Lau, "Formresnet: Formatted residual learning for image restoration," in *IEEE Conf. Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017, pp. 1034–1042.

[17] R. Vemulapalli, O. Tuzel, and M. Liu, "Deep Gaussian conditional random field network: A model-based deep network for discriminative denoising," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 4801–4809.

[18] T. Remez, O. Litany, R. Giryes, and A. M. Bronstein, "Deep class aware denoising," *arXiv preprint arXiv:1701.01698*, 2017.

[19] W. T. Freeman, T. Jones, and E. C. Pasztor, "Example-based super-resolution," *IEEE Computer Graphics and Applications*, vol. 22, no. 2, pp. 56–65, 2002.

[20] M. Zontak and M. Irani, "Internal statistics of a single natural image," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2011, pp. 977–984.

[21] A. Adams, N. Gelfand, J. Dolson, and M. Levoy, "Gaussian kd-trees for fast high-dimensional filtering," in *ACM Trans. on Graphics (ToG)*, vol. 28, no. 3, 2009.

[22] J. Hays and A. Efros, "Scene completion using millions of photographs," in *ACM Trans. on Graphics (TOG)*, vol. 26, no. 3, 2007, p. 4.

[23] S. H. Chan, T. Zickler, and Y. M. Lu, "Monte Carlo non-local means random sampling for large-scale image filtering," *IEEE Trans. on Image Processing*, vol. 23, no. 8, pp. 3711–3725, 2014.

[24] A. Levin and B. Nadler, "Natural image denoising: Optimality and inherent bounds," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conf. on*. IEEE, 2011, pp. 2833–2840.

[25] A. Levin, B. Nadler, F. Durand, and W. T. Freeman, "Patch complexity, finite pixel correlations and optimal denoising," in *European Conf. on Computer Vision (ECCV)*, 2012, pp. 73–86.

[26] S. Pyatykh and J. Hesser, "MMSE estimation for Poisson noise removal in images," *arXiv preprint arXiv:1512.00717*, 2015.

[27] K. He and J. Sun, "Computing nearest-neighbor fields via propagation-assisted kd-trees," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 111–118.

[28] I. Olonetsky and S. Avidan, "Treecann-kd tree coherence approximate nearest neighbor algorithm," *European Conf. on Computer Vision (ECCV)*, pp. 602–615, 2012.

[29] C. Barnes, E. Shechtman, D. B. Goldman, and A. Finkelstein, "The generalized patchmatch correspondence algorithm," in *European Conf. on Computer Vision (ECCV)*, 2010, pp. 29–43.

[30] E. Luo, S. Chan, and T. Nguyen, "Adaptive image denoising by targeted databases," *IEEE Trans. on Image Processing*, vol. 24, no. 7, pp. 2167–2181, 2015.

[31] S. Anwar, F. Porikli, and C. P. Huynh, "Category-specific object image denoising," *IEEE Trans. on Image Processing*, vol. 26, no. 11, pp. 5506–5518, 2017.

[32] A. Teodoro, J. Bioucas-Dias, and M. Figueiredo, "Image restoration and reconstruction using variable splitting and class-adapted image priors," in *IEEE International Conf. on Image Processing*, 2016.

[33] C. P. Robert, *Monte Carlo Methods*. Wiley, 2004.

[34] M. Evans and T. Swartz, "Methods for approximating integrals in statistics with special emphasis on Bayesian integration problems," *Statistical Science*, pp. 254–272, 1995.

[35] L. Wasserman, *All of statistics: a concise course in statistical inference*. Springer, 2013.

[36] T. C. Hesterberg, "Advances in importance sampling," Ph.D. dissertation, Stanford University, 1988.

[37] A. Owen, *Monte Carlo Theory, Methods, and Examples*. Unpublished, 2013, available at http://statweb.stanford.edu/~owen/mc/.

[38] H. Kahn and A. W. Marshall, "Methods of reducing sample size in Monte Carlo computations," *Journal of the Operations Research Society of America*, vol. 1, no. 5, pp. 263–278, 1953.

[39] T. Hesterberg, "Weighted average importance sampling and defensive mixture distributions," *Technometrics*, vol. 37, no. 2, pp. 185–194, 1995.

[40] O. Cappé, A. Guillin, J. Marin, and C. P. Robert, "Population Monte Carlo," *Journal of Computational and Graphical Statistics*, vol. 13, no. 4, pp. 907–929, 2004.

[41] J. Cornuet, J. Marin, A. Mira, and C. P. Robert, "Adaptive multiple importance sampling," *Scandinavian Journal of Statistics*, vol. 39, no. 4, pp. 798–812, 2012.

[42] M. Niknejad, J. Bioucas-Dias, and M. A. T. Figueiredo, "Class-specific image denoising using importance sampling," in *IEEE Int. Conf. on Image Processing (ICIP)*, 2017, pp. 1242–1246.

[43] ——, "Class-specific poisson denoising by patch-based importance sampling," in *IEEE Int. Conf. on Image Processing (ICIP)*, Sept. 2017, pp. 1247–1251.

[44] S. Kullback and R. A. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[45] L. Le Cam and G. L. Yang, *Asymptotics in statistics: some basic concepts*. Springer, 2012.

[46] A. Bhattacharyya, "On a measure of divergence between two statistical populations defined by their probability distribution," *Bull. Calcutta Math. Soc*, 1943.

[47] G. Celeux and G. Govaert, "A classification EM algorithm for clustering and two stochastic versions," *Computational Statistics and Data Analysis*, vol. 14, no. 3, pp. 315–332, 1992.

[48] E. Koblents and J. Míguez, "A population Monte Carlo scheme with transformed weights and its application to stochastic kinetic models," *Statistics and Computing*, vol. 25, no. 2, pp. 407–425, 2015.

[49] A. Foi, M. Trimeche, V. Katkovnik, and K. Egiazarian, "Practical Poissonian-Gaussian noise modeling and fitting for single-image raw-data," *IEEE Trans. on Image Processing*, vol. 17, no. 10, pp. 1737–1754, 2008.

[50] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma, "RASL: Robust alignment by sparse and low-rank decomposition for linearly correlated images," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2233–2246, 2012.

[51] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[52] A. Rond, R. Giryes, and M. Elad, "Poisson inverse problems by the plug-and-play scheme," *Journal of Visual Communication and Image Representation*, vol. 41, pp. 96–108, 2016.

[53] T. Remez, O. Litany, R. Giryes, and A. M. Bronstein, "Deep convolutional denoising of low-light images," *arXiv preprint arXiv:1701.01687*, 2017.

[54] H. Takeda, S. Farsiu, and P. Milanfar, "Kernel regression for image processing and reconstruction," *IEEE Trans. on Image Processing*, vol. 16, no. 2, pp. 349–366, 2007.

[55] S. Roth and M. J. Black, "Fields of experts: A framework for learning image priors," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 2005, pp. 860–867.

[56] S. H. Chan, T. Zickler, and Y. M. Lu, "Monte Carlo non-local means: Random sampling for large-scale image filtering," *IEEE Trans. on Image Processing*, vol. 23, no. 8, pp. 3711–3725, 2014.

[57] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *International Conf. on Computer Vision (ICCV)*, vol. 2, 2001, pp. 416–423.

1057-7149 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.