J Math Imaging Vis (2014) 48:279–294
DOI 10.1007/s10851-013-0435-6

Poisson Noise Reduction with Non-local PCA

Joseph Salmon · Zachary Harmany · Charles-Alban Deledalle · Rebecca Willett

Published online: 6 April 2013

Abstract Photon-limited imaging arises when the number of photons collected by a sensor array is small relative to the number of detector elements. Photon limitations are an important concern for many applications such as spectral imaging, night vision, nuclear medicine, and astronomy. Typically a Poisson distribution is used to model these observations, and the inherent heteroscedasticity of the data combined with standard noise removal methods yields significant artifacts. This paper introduces a novel denoising algorithm for photon-limited images which combines elements of dictionary learning and sparse patch-based representations of images. The method employs both an adaptation of Principal Component Analysis (PCA) for Poisson noise and recently developed sparsity-regularized convex optimization algorithms for photon-limited images. A comprehensive empirical evaluation of the proposed method helps characterize the performance of this approach relative to other state-of-the-art denoising methods. The results reveal that, despite its conceptual simplicity, Poisson PCA-based denoising appears to be highly competitive in very low light regimes.

Keywords Image denoising · PCA · Gradient methods · Newton's method · Signal representations

J. Salmon (✉)
Department LTCI, CNRS UMR 5141, Institut Mines-Télécom, Télécom ParisTech, Paris, France
e-mail: joseph.salmon@telecom-paristech.fr

Z. Harmany
Department of Electrical and Computer Engineering, University of Wisconsin-Madison, Madison, WI, USA
e-mail: harmany@wisc.edu

C.-A. Deledalle
IMB, CNRS-Université Bordeaux 1, Talence, France
e-mail: charles-alban.deledalle@math.u-bordeaux1.fr

R. Willett
Department of Electrical and Computer Engineering, Duke University, Durham, NC, USA
e-mail: willett@duke.edu

1 Introduction, Model, and Notation

In a broad range of imaging applications, observations correspond to counts of photons hitting a detector array, and these counts can be very small. For instance, in night vision, infrared, and certain astronomical imaging systems, there is a limited amount of available light. Photon limitations can even arise in well-lit environments when using a spectral imager which characterizes the wavelength of each received photon. The spectral imager produces a three-dimensional data cube, where each voxel in this cube represents the light intensity at a corresponding spatial location and wavelength. As the spectral resolution of these systems increases, the number of available photons for each spectral band decreases. Photon-limited imaging algorithms are designed to estimate the underlying spatial or spatio-spectral intensity underlying the observed photon counts.

There exists a rich literature on image estimation or denoising methods, and a wide variety of effective tools. The photon-limited image estimation problem is particularly challenging because the limited number of available photons introduces intensity-dependent Poisson statistics which require specialized algorithms and analysis for optimal performance. Challenges associated with low photon count data are often circumvented in hardware by designing systems which aggregate photons into fixed bins across space and wavelength (*i.e.*, creating low-resolution cameras). If the

bins are large enough, the resulting low spatial and spectral resolution cannot be overcome. High-resolution observations, in contrast, exhibit significant non-Gaussian noise since each pixel is generally either one or zero (corresponding to whether or not a photon is counted by the detector), and conventional algorithms which neglect the effects of photon noise will fail. Simply transforming Poisson data to produce data with approximate Gaussian noise (via, for instance, the variance stabilizing Anscombe transform [2, 31] or Fisz transform [17, 18]) can be effective when the number photon counts are uniformly high [5, 46]. However, when photon counts are very low these approaches may suffer, as shown later in this paper.

This paper demonstrates how advances in low-dimensional modeling and sparse Poisson intensity reconstruction algorithms can lead to significant gains in photon-limited (spectral) image accuracy at the resolution limit. The proposed method combines Poisson Principal Component Analysis (Poisson-PCA—a special case of the Exponential-PCA [10, 40]) and sparse Poisson intensity estimation methods [20] in a non-local estimation framework. We detail the targeted optimization problem which incorporates the heteroscedastic nature of the observations and present results improving upon state-of-the-art methods when the noise level is particularly high. We coin our method Poisson Non-Local Principal Component Analysis (Poisson NLPCA).

Since the introduction of non-local methods for image denoising [8], these methods have proved to outperform previously considered approaches [1, 11, 12, 30] (extensive comparisons of recent denoising method can be found for Gaussian noise in [21, 26]). Our work is inspired by recent methods combining PCA with patch-based approaches [15, 33, 47] for the Additive White Gaussian Noise (AWGN) model, with natural extensions to spectral imaging [13]. A major difference between these approaches and our method is that we directly handle the Poisson structure of the noise, without any "Gaussianization" of the data. Since our method does not use a quadratic data fidelity term, the singular value decomposition (SVD) cannot be used to solve the minimization. Our direct approach is particularly relevant when the image suffers from a high noise level (*i.e.*, low photon emissions).

1.1 Organization of the Paper

In Sect. 1.2, we describe the mathematical framework. In Sect. 2, we recall relevant basic properties of the exponential family, and propose an optimization formulation for matrix factorization. Section 3 provides an algorithm to iteratively compute the solution of our minimization problem. In Sect. 5, an important clustering step is introduced both to improve the performance and the computational complexity of our algorithm. Algorithmic details and experiments are reported in Sects. 6 and 7, and we conclude in Sect. 8.

1.2 Problem Formulation

For an integer $M > 0$, the set $\{1, \ldots, M\}$ is denoted $[\![1, M]\!]$. For $i \in [\![1, M]\!]$, let y_i be the observed pixel values obtained through an image acquisition device. We consider each y_i to be an independent random Poisson variable whose mean $f_i \geq 0$ is the underlying intensity value to be estimated. Explicitly, the discrete Poisson probability of each y_i is

$$\mathbb{P}(y_i \,|\, f_i) = \frac{f_i^{y_i}}{y_i!} e^{-f_i}, \tag{1}$$

where 0! is understood to be 1 and 0^0 to be 1.

A crucial property of natural images is their ability to be accurately represented using a concatenation of patches, each of which is a simple linear combination of a small number of representative atoms. One interpretation of this property is that the patch representation exploits self-similarity present in many images, as described in AWGN settings [11, 12, 30]. Let Y denote the $M \times N$ matrix of all the vectorized $\sqrt{N} \times \sqrt{N}$ overlapping patches (neglecting border issues) extracted from the noisy image, and let F be defined similarly for the true underlying intensity. Thus $Y_{i,j}$ is the jth pixel in the ith patch.

Many methods have been proposed to represent the collection of patches in a low dimensional space in the same spirit as PCA. We use the framework considered in [10, 40], that deals with data well-approximated by random variables drawn from exponential family distributions. In particular, we use Poisson-PCA, which we briefly introduce here before giving more details in the next section. With Poisson-PCA, one aims to approximate F by:

$$F_{i,j} \approx \exp\big([UV]_{i,j}\big) \quad \forall (i, j) \in [\![1, M]\!] \times [\![1, N]\!], \tag{2}$$

where

- U is the $M \times \ell$ matrix of coefficients;
- V is the $\ell \times N$ matrix representing the dictionary components or axis. The rows of V represents the dictionary elements; and
- $\exp(UV)$ is the element-wise exponentiation of UV: $\exp([UV]_{i,j}) := [\exp(UV)]_{i,j}$.

The approximation in (2) is different than the approximation model used in similar methods based on AWGN, where typically one assumes $F_{i,j} \approx [UV]_{i,j}$ (that is, without exponentiation). Our exponential model allows us to circumvent challenging issues related to the nonnegativity of F and thus facilitates significantly faster algorithms.

The goal is to compute an estimate of the form (2) from the noisy patches Y. We assume that this approximation is accurate for $\ell \ll M$, whereby restricting the rank ℓ acts to regularize the solution. In the following section we elaborate on this low-dimensional representation.

2 Exponential Family and Matrix Factorization

We present here the general case of matrix factorization for an exponential family, though in practice we only use this framework for the Poisson and Gaussian cases. We describe the idea for a general exponential family because our proposed method considers Poisson noise, but we also develop an analogous method (for comparison purposes) based on an Anscombe transform of the data and a Gaussian noise model. The solution we focus on follows the one introduced by [10]. Some more specific details can be found in [39, 40] about matrix factorization for exponential families.

2.1 Background on the Exponential Family

We assume that the observation space $\mathcal{Y}$ is equipped with a σ-algebra $\mathcal{B}$ and a dominating σ-finite measure ν on $(\mathcal{Y}, \mathcal{B})$. Given a positive integer n, let ϕ: $\mathcal{Y} \to \mathbb{R}^n$ be a measurable function, and let ϕ_k, $k = 1, 2, \ldots, n$ denote its components: $\phi(y) = (\phi_1(y), \ldots, \phi_n(y))$.

Let Θ be defined as the set of all $\theta \in \mathbb{R}^n$ such that $\int_{\mathcal{Y}} \exp(\langle \theta | \phi(y) \rangle) d\nu < \infty$. We assume it is convex and open in this paper. We then have the following definition:

Definition 1 An *exponential family with sufficient statistic* ϕ is the set $\mathcal{P}(\phi)$ of probability distributions w.r.t. the measure ν on $(\mathcal{Y}, \mathcal{B})$ parametrized by $\theta \in \Theta$, such that each probability density function $p_\theta \in \mathcal{P}(\phi)$ can be expressed as

$$p_\theta(y) = \exp\big\{\big\langle \theta | \phi(y) \big\rangle - \Phi(\theta)\big\}, \tag{3}$$

where

$$\Phi(\theta) = \log \int_{\mathcal{Y}} \exp\big\{\big\langle \theta | \phi(y) \big\rangle\big\} d\nu(y). \tag{4}$$

The parameter $\theta \in \Theta$ is called the *natural parameter* of $\mathcal{P}(\phi)$, and the set Θ is called the *natural parameter space*. The function Φ is called the *log partition function*. We denote by $\mathbb{E}_\theta[\cdot]$ the expectation w.r.t. p_θ:

$$\mathbb{E}_\theta\big[g(X)\big] = \int_{\mathcal{X}} g(y)\big(\exp\big(\big\langle \theta | \phi(y) \big\rangle\big) - \Phi(\theta)\big) d\nu(y).$$

Example 1 Assume the data are independent (not necessarily identically distributed) Gaussian random variables with means μ_i and (known) variances σ^2. Then the parameters are: $\forall y \in \mathbb{R}^n, \phi(y) = y$, $\Phi(\theta) = \sum_{i=1}^n \theta_i^2 / 2\sigma^2$ and $\nabla\Phi(\theta) = (\theta_1/\sigma^2, \ldots, \theta_n/\sigma^2)$ and ν is the Lebesgue measure on $\mathbb{R}^n$ (*cf.* [34] for more details on the Gaussian distribution, possibly with non-diagonal covariance matrix).

Example 2 For Poisson distributed data (not necessarily identically distributed), the parameters are the following: $\forall y \in \mathbb{R}^n, \phi(y) = y$, and $\Phi(\theta) = \langle \exp(\theta) | \mathbb{1}_n \rangle = \sum_{i=1}^n e^{\theta_i}$, where exp is the component-wise exponential function:

$$\exp : (\theta_1, \ldots, \theta_n) \mapsto \big(e^{\theta_1}, \ldots, e^{\theta_n}\big), \tag{5}$$

and $\mathbb{1}_n$ is the vector $(1, \ldots, 1)^\top \in \mathbb{R}^n$. Moreover $\nabla\Phi(\theta) = \exp(\theta)$ and ν is the counting measure on $\mathbb{N}$ weighted by $e/n!$.

Remark 1 The standard parametrization is usually different for Poisson distributed data, and this family is often parametrized by the rate parameter $f = \exp(\theta)$.

2.2 Bregman Divergence

The general measure of proximity we use in our analysis relies on Bregman divergence [7]. For exponential families, the relative entropy (Kullback-Leibler divergence) between p_{θ_1} and p_{θ_2} in $\mathcal{P}(\phi)$, defined as

$$D_\Phi(p_{\theta_1} || p_{\theta_2}) = \int_{\mathcal{X}} p_{\theta_1} \log(p_{\theta_1} / p_{\theta_2}) d\nu, \tag{6}$$

can be simply written as a function of the natural parameters:

$$D_\Phi(p_{\theta_1} || p_{\theta_2}) = \Phi(\theta_2) - \Phi(\theta_1) - \big\langle \nabla\Phi(\theta_1) | \theta_2 - \theta_1 \big\rangle.$$

From the last equation, we have that the mapping $D_\Phi : \Theta \times \Theta \to \mathbb{R}$, defined by $D_\Phi(\theta_1, \theta_2) = D_\Phi(p_{\theta_2} || p_{\theta_2})$, is a Bregman divergence.

Example 3 For Gaussian distributed observations with unit variance and zero mean, the Bregman divergence can be written:

$$D_G(\theta_1, \theta_2) = \|\theta_1 - \theta_2\|_2^2. \tag{7}$$

Example 4 For Poisson distributed observations, the Bregman divergence can be written:

$$D_P(\theta_1, \theta_2) = \big\langle \exp(\theta_2) - \exp(\theta_1) | \mathbb{1}_n \big\rangle - \big\langle \exp(\theta_1) | \theta_2 - \theta_1 \big\rangle. \tag{8}$$

We define the matrix Bregman divergence as

$$D_\Phi(X || Y) = \Phi(Y) - \Phi(X) - \mathrm{Tr}\big(\big(\nabla\Phi(X)\big)^\top (X - Y)\big), \tag{9}$$

for any (non necessarily square) matrices X and Y of size $M \times N$.

2.3 Matrix Factorization and Dictionary Learning

Suppose that one observes $Y \in \mathbb{R}^{M \times N}$, and let $Y_{i,:}$ denote the ith patch in row-vector form. We would like to approximate the underlying intensity F by a combination of some vectors, atoms, or dictionary elements $V = [v_1, \ldots, v_\ell]$,

where each patch uses different weights on the dictionary elements. In other words, the ith patch of the true intensity, denoted $F_{i,:}$, is approximated as $\exp(u_i V)$, where u_i is the ith row of U and contains the dictionary weights for the ith patch. Note that we perform this factorization in the natural parameter space, which is why we use the exponential function in the formulation given in Eq. (2).

Using the divergence defined in (9) our objective is to find U and V minimizing the following criterion:

$$D_\Phi(Y||UV) = \sum_{j=1}^{M} \Phi(u_j V) - Y_{j,:} - \langle Y_{j,:} | u_j V - Y_{j,:} \rangle.$$

In the Poisson case, the framework introduced in [10, 40] uses the Bregman divergence in Example 4 and amounts to minimizing the following loss function

$$L(U,V) = \sum_{i=1}^{M}\sum_{j=1}^{N} \exp(UV)_{i,j} - Y_{i,j}(UV)_{i,j} \tag{10}$$

with respect to the matrices U and V. Defining the corresponding minimizers of the biconvex problem

$$\big(U^*, V^*\big) \in \underset{(U,V)\in\mathbb{R}^{M\times\ell}\times\mathbb{R}^{\ell\times N}}{\arg\min} L(U,V), \tag{11}$$

our image intensity estimate is

$$\widehat{F} = \exp\big(U^* V^*\big). \tag{12}$$

This is what we call Poisson-PCA (of order ℓ) in the remainder of the paper.

Remark 2 The classical PCA (of order ℓ) is obtained using the Gaussian distribution, which leads to solving the same minimization as in Eq. (11), except that L is replaced by

$$\tilde{L}(U,V) = \sum_{i=1}^{M}\sum_{j=1}^{N} \big((UV)_{i,j} - Y_{i,j}\big)^2.$$

Remark 3 The problem as stated is non-identifiable, as scaling the dictionary elements and applying an inverse scaling to the coefficients would result in an equivalent intensity estimate. Thus, one should normalize the dictionary elements so that the coefficients cannot be too large and create numerical instabilities. The easiest solution is to impose that the atoms v_i are normalized w.r.t. the standard Euclidean norm, *i.e.*, for all $i \in \{1,\dots,\ell\}$ one ensures that the constraint $\|v_i\|_2^2 = \sum_{j=1}^{n} V_{i,j}^2 = 1$ is satisfied. In practice though, relaxing this constraint modifies the final output in a negligible way while helping to keep the computational complexity low.

3 Newton's Method for Minimizing L

Here we follow the approach proposed by [19, 35] that consists in using Newton steps to minimize the function L. Though L is not jointly convex in U and V, when fixing one variable and keeping the other fixed the partial optimization problem is convex (*i.e.*, the problem is biconvex). Therefore we consider Newton updates on the partial problems. To apply Newton's method, one needs to invert the Hessian matrices with respect to both U and V, defined by $H_U = \nabla_U^2 L(U,V)$ and $H_V = \nabla_V^2 L(U,V)$. Simple algebra leads to the following closed form expressions for the components of these matrices (for notational simplicity we use pixel coordinates to index the entries of the Hessian):

$$\frac{\partial^2 L(U,V)}{\partial U_{a,b}\partial U_{c,d}} = \begin{cases} \sum_{j=1}^{N} \exp(UV)_{a,j} V_{b,j}^2, & \text{if } (a,b)=(c,d), \\ 0 & \text{otherwise,} \end{cases}$$

and

$$\frac{\partial^2 L(U,V)}{\partial V_{a,b}\partial V_{c,d}} = \begin{cases} \sum_{i=1}^{M} U_{i,a}^2 \exp(UV)_{i,b}, & \text{if } (a,b)=(c,d), \\ 0 & \text{otherwise,} \end{cases}$$

where both partial Hessians can be represented as diagonal matrices (*cf.* Appendix C for more details).

We propose to update the rows of U and columns of V as proposed in [35]. We introduce the function Vect_C that transforms a matrix into one single column (concatenates the columns), and the function Vect_R that transforms a matrix into a single row (concatenates the rows). Precise definitions are given in Appendix D. The updating step for U and V are then given respectively by

$$\mathrm{Vect}_R(U_{t+1}) = \mathrm{Vect}_R(U_t) - \mathrm{Vect}_R\big(\nabla_U L(U_t,V_t)\big) H_{U_t}^{-1}$$

and

$$\mathrm{Vect}_C(V_{t+1}) = \mathrm{Vect}_C(V_t) - H_{V_t}^{-1}\,\mathrm{Vect}_C\big(\nabla_V L(U_t,V_t)\big).$$

Simple algebra (*cf.* Appendix D or [19] for more details) leads to the following updating rules for the ith row of U_{t+1} (denoted $U_{t+1,i,:}$):

$$U_{t+1,i,:} = U_{t,i,:} - \big(\exp(U_t V_t)_{i,:} - Y_{i,:}\big) V_t^\top \big(V_t D_i V_t^\top\big)^{-1}, \tag{13}$$

where $D_i = \mathrm{diag}(\exp(U_t V_t)_{i,1},\dots,\exp(U_t V_t)_{i,N})$ is a diagonal matrix of size $N\times N$. The updating rule for $V_{t,:,j}$, the jth column of V_t, is computed in a similar way, leading to

$$V_{t+1,:,j} = V_{t,:,j} - \big(U_{t+1}^\top E_j U_{t+1}\big)^{-1} U_{t+1}^\top \big(\exp(U_{t+1}V_t)_{:,j} - Y_{:,j}\big), \tag{14}$$

Algorithm 1 Poisson NLPCA/ NLSPCA

Inputs: Noisy pixels y_i for $i = 1, \ldots, M$
Parameters: Patch size $\sqrt{N} \times \sqrt{N}$, number of clusters K, number of components ℓ, maximal number of iterations N_{iter}
Output: estimated image $\widehat{f}$
Method:
Patchization: create the collection of patches for the noisy image Y
Clustering: create K clusters of patches using K-Means
The kth cluster (represented by a matrix Y^k) has M_k elements
for all cluster k **do**
  Initialize $U_0 = \text{randn}(M_k, \ell)$ and $V_0 = \text{randn}(\ell, N)$
  while $t \leq N_{\text{iter}}$ and test $> \varepsilon_{\text{stop}}$ **do**
    for all $i \leq M_k$ **do**
      Update the ith row of U using (13) or (17)–(19)
    end for
    for all $j \leq \ell$ **do**
      Update the jth column of V using (3)
    end for
    $t := t + 1$
  end while
  $\widehat{F}^k = \exp(U_t V_t)$
end for
Concatenation: fuse the collection of denoised patches $\widehat{F}$
Reprojection: average the various pixel estimates due to overlaps to get an image estimate: $\widehat{f}$

where $E_j = \text{diag}(\exp(U_{t+1}V_t)_{1,j}, \ldots, \exp(U_{t+1}V_t)_{M,j})$ is a diagonal matrix of size $M \times M$. More details about the implementation are given in Algorithm 1.

4 Improvements Through ℓ_1 Penalization

A possible alternative to minimizing Eq. (10), consists of minimizing a penalized version of this loss, whereby a sparsity constraint is imposed on the elements of U (the dictionary coefficients). Related ideas have been proposed in the context of sparse PCA [48], dictionary learning [27], and matrix factorization [29, 30] in the Gaussian case. Specifically, we minimize

$$L^{\text{Pen}}(U, V) = L(U, V) + \lambda \,\text{Pen}(U), \tag{15}$$

where $\text{Pen}(U)$ is a penalty term that ensures we use only a few dictionary elements to represent each patch. The parameter λ controls the trade-off between data fitting and sparsity. We focus on the following penalty function:

$$\text{Pen}(U) = \sum_{i,j} |U_{i,j}|. \tag{16}$$

We refer to the method as the Poisson Non-Local Sparse PCA (NLSPCA).

The algorithm proposed in [29] can be adapted with the SpaRSA step provided in [44], or in our setting by using its adaptation to the Poisson case—SPIRAL [20]. First one should note that the updating rule for the dictionary element, *i.e.*, Eq. (3), is not modified. Only the coefficient update, *i.e.*, Eq. (13) is modified as follows:

$$U_{t+1,:} = \arg\min_{u \in \mathbb{R}^\ell} \langle \exp(uV_t) | \mathbb{1} \rangle - \langle uV_t | Y_{t+1,:} \rangle + \lambda \|u\|_1. \tag{17}$$

For this step, we use the SPIRAL approach. This leads to the following updating rule for the coefficients:

$$\begin{aligned} U_{t+1,:} = \arg\min_{z \in \mathbb{R}^\ell} \quad & \frac{1}{2}\|z - \gamma_t\|_2^2 + \frac{\lambda}{\alpha_t}\|z\|_1, \\ \text{subject to} \quad & \gamma_t = U_{t,:} - \frac{1}{\alpha_t}\nabla_U f(U_{t,:}), \end{aligned} \tag{18}$$

where $\alpha_t > 0$ and the function f is defined by

$$f(u) = \langle \exp(uV_t) | \mathbb{1} \rangle - \langle uV_t | Y_{t+1,:} \rangle.$$

The gradient can thus be expressed as

$$\nabla f(u) = \big(\exp(uV_{t+1}) - Y_{t+1,:}\big) V_{t+1}^\top.$$

Then the solution of the problem (18), is simply

$$U_{t+1,:} = \eta_{\text{ST}}\left(\gamma_t, \frac{\lambda}{\alpha_t}\right), \tag{19}$$

where η_{ST} is the soft-thresholding function $\eta_{\text{ST}}(x, \tau) = \text{sign}(x) \cdot (|x| - \tau)_+$.

Other methods than SPIRAL for solving the Poisson ℓ_1-constrained problem could be investigated, *e.g.*, Alternating Direction Method of Multipliers (ADMM) algorithms for ℓ_1-minimization (*cf.* [6, 45], or one specifically adapted to Poisson noise [16]), though choosing the augmented Lagrangian parameter for these methods can be challenging in practice.

5 Clustering Step

Most strategies apply matrix factorization on patches extracted from the entire image. A finer strategy consists in first performing a clustering step, and then applying matrix factorization on each cluster. Indeed, this avoids grouping dissimilar patches of the image, and allows us to represent the data within each cluster with a lower dimensional dictionary. This may also improve on the computation time of the dictionary. In [11, 30], the clustering is based on a geometric partitioning of the image. This improves on the global approach but may results in poor estimation where the partition is too small. Moreover, this approach remains local and cannot exploit the redundancy inside similar disconnected regions. We suggest here using a non-local approach where

the clustering is directly performed in the patch domain similarly to [9]. Enforcing similarity inside non-local groups of patches results in a more robust low rank representation of the data, decreasing the size of the matrices to be factorized, and leading to efficient algorithms. Note that in [15], the authors studied an hybrid approach where the clustering is driven in a hierarchical image domain as well as in the patch domain to provide both robustness and spatial adaptivity. We have not considered this approach since, while increasing the computation load, it yields to significant improvements particularly at low noise levels, which are not the main focus of this paper.

For clustering we have compared two solutions: one using only a simple K-means on the original data, and one performing a Poisson K-means. In similar fashion for adapting PCA for exponential families, the K-means clustering algorithm can also be generalized using Bregman divergences; this is called Bregman clustering [3]. This approach, detailed in Algorithm 2, has an EM (Expectation-Maximization) flavor and is proved to converge in a finite number of steps.

The two variants we consider differ only in the choice of the divergence d used to compare elements x with respect to the centers of the clusters x_C:

- Gaussian: Uses the divergence defined in (7):

$$d(f, f_C) = D_G(f, f_C) = \|f - f_C\|_2^2.$$

- Poisson: Uses the divergence defined in (8):

$$d(f, f_C) = D_P\big(\log(f), \log(f_C)\big) = \sum_j f_C^j - f^j \log\big(f_C^j\big)$$

  where the log is understood element-wise (note that the difference with (8) is only due to a different parametrization here).

In our experiments, we have used a small number (for instance $K = 14$) of clusters fixed in advance.

In the low-intensity setting we are targeting, clustering on the raw data may yield poor results. A preliminary image estimate might be used for performing the clustering, especially if one has a fast method giving a satisfying denoised image. For instance, one can apply the Bregman hard clustering on the denoised images obtained after having performed the full Poisson NLPCA on the noisy data. This approach was the one considered in the short version of this paper [36], where we were using only the classical K-means. However, we have noticed that using the Poisson K-means instead leads to a significant improvement. Thus, the benefit of iterating the clustering is lowered. In this version, we do not consider such iterative refinement of the clustering. The entire algorithm is summarized in Fig. 1.

Algorithm 2 Bregman hard clustering

Inputs: Data points: $(f_i)_{i=1}^M \in \mathbb{R}^N$, number of clusters: K, Bregman divergence: $d : \mathbb{R}^N \times \mathbb{R}^N \mapsto \mathbb{R}^+$
Output: Clusters centers: $(\mu_k)_{k=1}^K$, partition associated : $(\mathcal{C}_k)_{k=1}^K$
Method:
Initialize $(\mu_k)_{k=1}^K$ by randomly selecting K elements among $(f_i)_{i=1}^M$
repeat
  (*The Assignment step: Cluster updates*)
  Set $\mathcal{C}_k := \emptyset$, $1 \le k \le K$
  for $i = 1, \ldots, M$ **do**
    $\mathcal{C}_{k^*} := \mathcal{C}_{k^*} \cup \{f_i\}$
    where $k^* = \arg\min_{k'=1,\ldots,K} d(f_i, \mu_{k'})$
  end for
  (*The Estimation step: Center updates*)
  for $k = 1, \ldots, K$ **do**
    $\mu_k := \frac{1}{\#\mathcal{C}_k} \sum_{f_i \in \mathcal{C}_k} f_i$
  end for
until convergence

6 Algorithmic Details

We now present the practical implementation of our method, for the two variants that are the Poisson NLPCA and the Poisson NLSPCA.

6.1 Initialization

We initialize the dictionary at random, drawing the entries from a standard normal distribution, that we then normalize to have a unit Euclidean norm. This is equivalent to generating the atoms uniformly at random from the Euclidean unit sphere. As a rule of thumb, we also constrain the first atom (or axis) to be initialized as a constant vector. However, this constraint is not enforced during the iterations, so this property can be lost after few steps.

6.2 Stopping Criterion and Conditioning Number

Many methods are proposed in [44] for the stopping criterion. Here we have used a criterion based on the relative change in the objective function $L^{\mathrm{Pen}}(U, V)$ defined in Eq. (15). This means that we iterate the alternating updates in the algorithm as long $\|\exp(U_t V_t) - \exp(U_{t+1} V_{t+1})\|^2 / \|\exp(U_t V_t)\|^2 \le \varepsilon_{\mathrm{stop}}$ for some (small) real number $\varepsilon_{\mathrm{stop}}$.

For numerical stability we have added a Tikhonov (or ridge) regularization term. Thus, we have substituted $V_t D_i V_t^\top$ in Eq. (13) with $(V_t D_i V_t^\top + \varepsilon_{\mathrm{cond}} I_\ell)$ and $(U_t^\top E_j U_t)$ in Eq. (3) with $(U_t^\top E_j U_t) + \varepsilon_{\mathrm{cond}} I_\ell$. For the NLSPCA version the $\varepsilon_{\mathrm{cond}}$ parameter is only used to update the dictionary in Eq. (3), since the regularization on the coefficients is provided by Eq. (17).

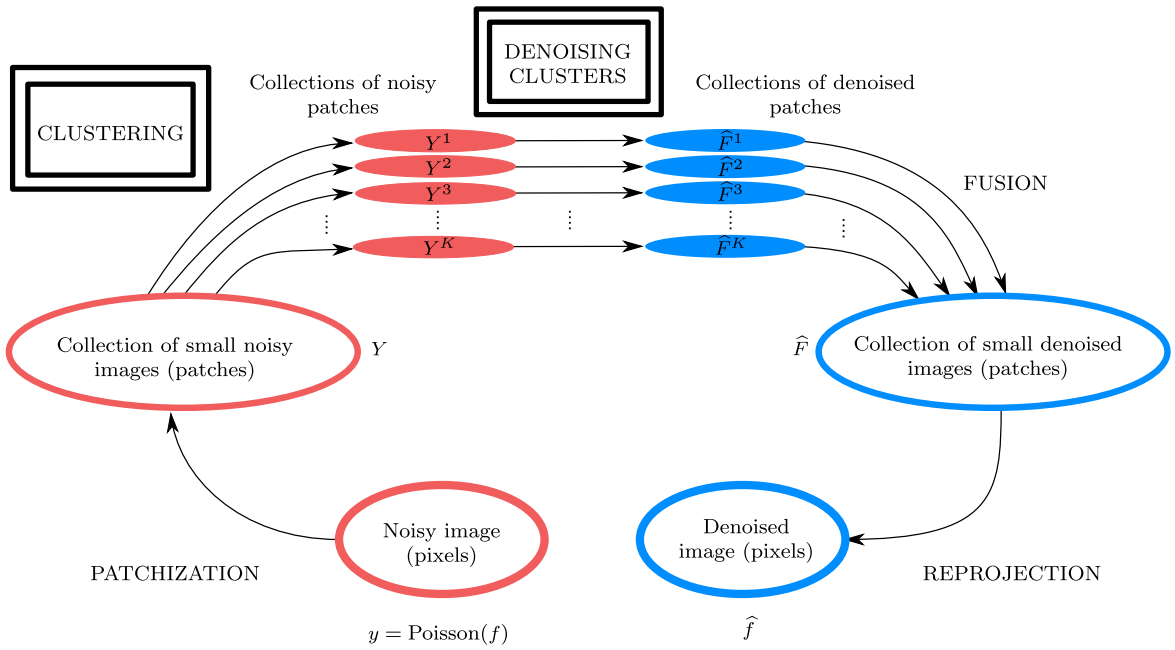


Fig. 1 Visual summary of our denoising method. In this work we mainly focus on the two highlighted points of the figure: **clustering** in the context of very photon-limited data, and specific **denoising** method for each cluster

6.3 Reprojections

Once the whole collection of patches is denoised, it remains to reproject the information onto the pixels. Among various solutions proposed in the literature (see for instance [37] and [11]) the most popular, the one we use in our experiments, is to uniformly average all the estimates provided by the patches containing the given pixel.

6.4 Binning-Interpolating

Following a suggestion of an anonymous reviewer, we have also investigated the following "binned" variant of our method:

1. aggregate the noisy Poisson pixels into small (for instance 3×3) bins, resulting in a smaller Poisson image with lower resolution but higher counts per pixel;
2. denoise this binned image using our proposed method;
3. enlarge the denoised image to the original size using (for instance bilinear) interpolation.

Indeed, in the extreme noise level case we have considered, this approach significantly reduces computation time, and for some images it yields a significant performance increase. The binning process allows us to implicitly use larger patches, without facing challenging memory and computation time issues. Of course, such a scheme could be applied to any method dealing with low photon counts, and we provide a comparison with the BM3D method (the best overall competing method) in the experiments section.

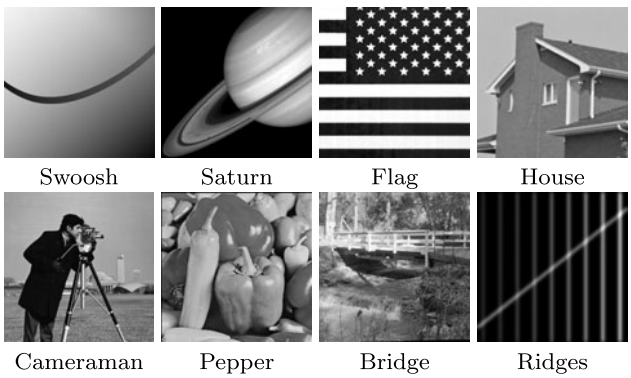


Fig. 2 Original images used for our simulations

7 Experiments

We have conducted experiments both on simulated and on real data, on grayscale images (2D) and on spectral images (3D). We summarize our results in the following, both with visual results and performance metrics.

7.1 Simulated 2D Data

We have first conducted comparisons of our method and several competing algorithms on simulated data. The images we have used in the simulations are presented in Fig. 2. We have considered the same noise level for the Saturn image (*cf.* Fig. 8) as in [41], where one can find extensive comparisons with a variety of multiscale methods [23, 24, 42].

Table 1 Parameter settings used in the proposed method. Note: M_k is the number of patches in the kth cluster as determined by the Bregman hard clustering step

Parameter	Definition	Value
N	patch size	20×20
ℓ	approximation rank	4
K	clusters	14
N_{iter}	iteration limit	20
$\varepsilon_{\text{stop}}$	stopping tolerance	10^{-1}
$\varepsilon_{\text{cond}}$	conditioning parameter	10^{-3}
λ	ℓ_1 regularization (NL-SPCA only)	$70\sqrt{\frac{\log(M_k)}{n}}$

In terms of PSNR, defined in the classical way (for 8-bit images)

$$\text{PSNR}(\widehat{f}, f) = 10 \log_{10} \frac{255^2}{\frac{1}{M}\sum_i (\widehat{f}_i - f_i)^2}, \tag{20}$$

our method globally improves upon other state-of-the-art methods such as Poisson-NLM [14], SAFIR [5], and Poisson Multiscale Partitioning (PMP) [42] for the very low light levels of interest. Moreover, visual artifacts tend to be reduced by our Poisson NLPCA and NLSPCA, with respect to the version using an Anscombe transform and classical PCA (*cf.* AnscombeNLPCA in Figs. 8 and 6 for instance). See Sect. 7.4 for more details on the methods used for comparison.

All our results for 2D and 3D images are provided for both the NLPCA and NLSPCA using (except otherwise stated) the parameter values summarized in Table 1. The step-size parameter α_t for the NL-SPCA method is chosen via a selection rule initialized with the Barzilai-Borwein choice, as described in [20].

7.2 Simulated 3D Data

In this section we have tested a generalization of our algorithm for spectral images. We have thus considered the NASA AVIRIS (Airborne Visible/Infrared Imaging Spectrometer) Moffett Field reflectance data set, and we have kept a $256 \times 256 \times 128$ sized portion of the total data cube. For the simulation we have used the same noise level as in [25] (the number of photons per voxel is 0.0387), so that comparison could be done with the results presented in this paper. Moreover to ease comparison with earlier work, the performance has been measured in terms of mean absolute error (MAE), defined by

$$\text{MAE}(\widehat{f}, f) = \frac{\|\widehat{f} - f\|_1}{\|f\|_1}. \tag{21}$$

We have performed the clustering on the 2D image obtained by summing the photons on the third (spectral) dimension, and using this clustering for each 3D patch. This approach is particularly well suited for low photons counts since with other approaches the clustering step can be of poor quality. Our approach provides an illustration of the importance of taking into account the correlations across the channels. We have used non-square patches since the spectral image intensity has different levels of homogeneity across the spectral and spatial dimensions. We thus have considered elongated patches with respect to the third dimension. In practice, the patch size used for the results presented is $5 \times 5 \times 23$, the number of clusters is $K = 30$, and the order of approximation is $\ell = 2$.

For the noise level considered, our proposed algorithm outperforms the other methods, BM4D [28] and PMP [25], both visually and in term of MAE (*cf.* Fig. 9). Again, these competing methods are described in Sect. 7.4.

7.3 Real 3D Data

We have also used our method to denoise some real noisy astronomical data. The last image we have considered is based on thermal X-ray emissions of the youngest supernova explosion ever observed. It is the supernova remnant G1.9+0.3 (@ NASA/CXC/SAO) in the Milky Way. The study of such spectral images can provide important information about the nature of elements present in the early stages of supernova. We refer to [4] for deeper insights on the implications for astronomical science. This dataset has an average of 0.0137 photons per voxel.

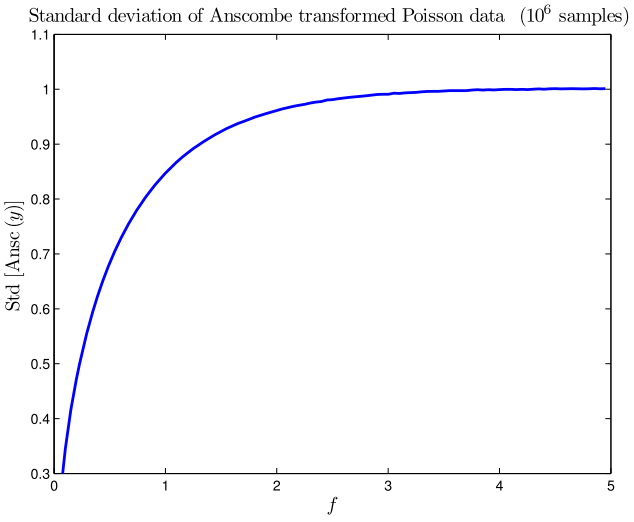
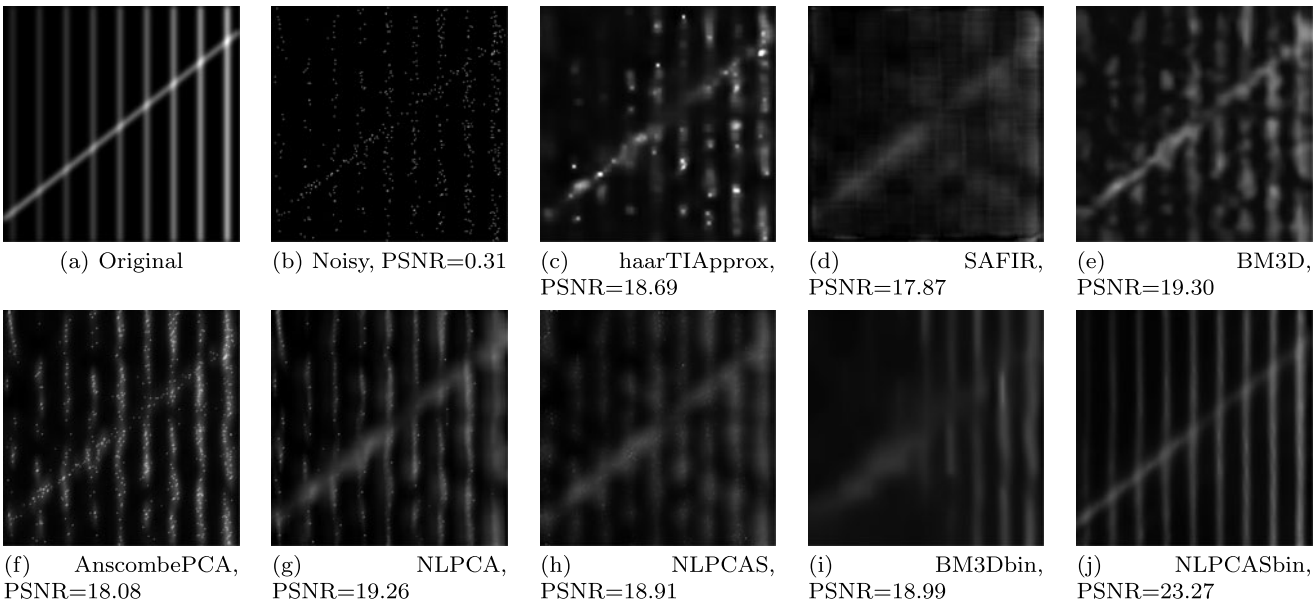


Fig. 3 Standard deviation approximation of some simulated Poisson data, after performing the Anscombe transform (Ansc). For each true parameter f, 10^6 Poisson realizations where drawn and the corresponding standard deviation is reported

Table 2 Experiments on simulated data (average over five noise realizations). Flag and Saturn images are displayed in Figs. 8, 6 and 7, and the others are given in [38] and in [46]

Method	Swoosh	Saturn	Flag	House	Cam	Pepper	Bridge	Ridges
				Peak = 0.1				
NLBayes	11.08	12.65	7.14	10.94	10.54	11.52	10.58	15.97
haarTIApprox	19.84	19.36	12.72	18.15	17.18	**19.10**	16.64	18.68
SAFIR	18.88	20.39	12.24	17.45	16.22	18.53	16.55	17.97
BM3D	17.21	19.13	13.12	16.63	15.75	17.24	15.72	19.47
BM3Dbin	**21.91**	**20.82**	14.36	18.39	17.11	18.84	**16.94**	20.33
NLPCA	19.12	20.40	14.45	18.06	16.58	18.48	16.48	21.25
NLSPCA	19.18	20.45	14.50	18.08	16.64	18.49	16.52	20.56
NLSPCAbin	21.56	19.47	**15.57**	**18.68**	**17.29**	18.73	16.90	**23.52**
				Peak = 0.2				
NLBayes	14.18	14.75	8.20	13.54	12.71	13.89	12.59	16.19
haarTIApprox	21.55	20.91	13.97	19.25	**18.37**	20.13	17.46	20.46
SAFIR	20.86	21.71	13.65	18.83	17.38	19.88	17.41	18.58
BM3D	20.27	21.20	14.25	18.67	17.44	19.31	17.14	21.10
BM3Dbin	**24.14**	**22.59**	16.04	**19.93**	18.24	**20.22**	**17.66**	23.92
NLPCA	21.20	22.29	16.53	19.08	17.80	19.69	17.49	24.10
NLSPCA	21.27	22.34	16.47	19.11	17.77	19.70	17.51	24.41
NLSPCAbin	24.04	20.56	**16.65**	19.87	17.90	19.61	17.43	**25.43**
				Peak = 0.5				
NLBayes	19.60	18.28	10.19	17.01	15.68	16.90	15.11	16.77
haarTIApprox	23.59	23.27	16.25	20.65	**19.59**	21.30	18.32	23.07
SAFIR	22.70	24.23	16.20	20.37	18.84	21.25	18.42	20.90
BM3D	23.53	24.09	15.94	20.50	18.86	21.03	18.37	23.33
BM3Dbin	26.20	**25.64**	18.53	**21.70**	19.58	**21.60**	**18.75**	27.99
NLPCA	24.50	25.38	**18.93**	20.78	19.36	21.13	18.47	**28.06**
NLSPCA	24.44	25.06	18.92	20.76	19.23	21.12	18.46	28.03
NLSPCAbin	**26.36**	20.67	17.09	20.97	18.39	20.28	18.16	26.81



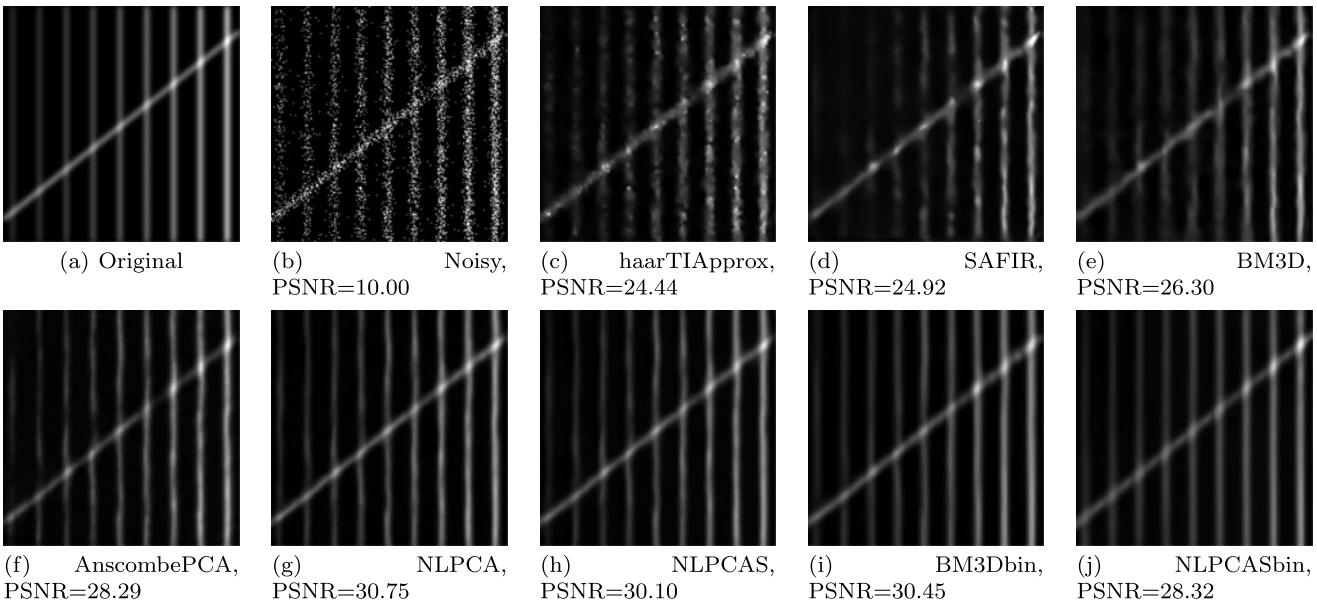
(a) Original (b) Noisy, PSNR=0.31 (c) haarTIApprox, PSNR=18.69 (d) SAFIR, PSNR=17.87 (e) BM3D, PSNR=19.30

(f) AnscombePCA, PSNR=18.08 (g) NLPCA, PSNR=19.26 (h) NLPCAS, PSNR=18.91 (i) BM3Dbin, PSNR=18.99 (j) NLPCASbin, PSNR=23.27

Fig. 4 Toy cartoon image (Ridges) corrupted with Poisson noise with Peak = 0.1

Table 2 (*Continued*)

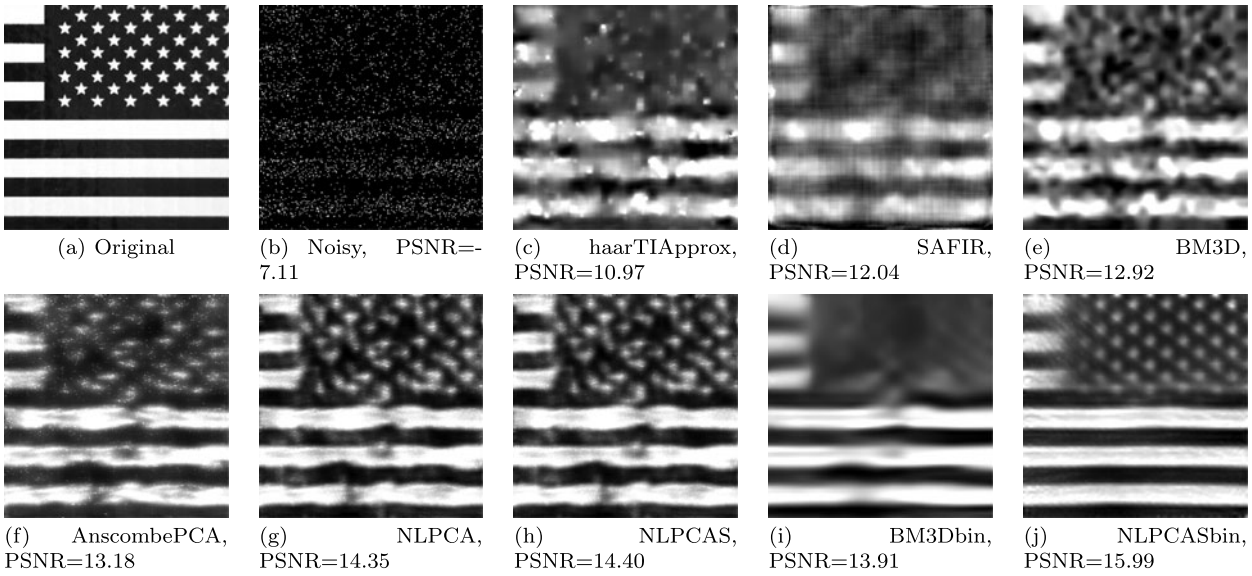
Method	Swoosh	Saturn	Flag	House	Cam	Pepper	Bridge	Ridges
	Peak = 1							
NLBayes	23.58	21.66	14.00	19.27	17.99	19.48	16.85	18.35
haarTIApprox	25.12	25.06	17.79	21.97	**20.64**	22.25	19.08	24.52
SAFIR	23.37	25.14	17.91	21.46	20.01	22.08	19.12	24.67
BM3D	26.21	25.88	18.45	22.26	20.45	22.27	19.39	25.76
BM3Dbin	**27.95**	**27.24**	19.49	**23.26**	20.61	**22.53**	**19.47**	29.91
NLPCA	26.99	27.08	20.23	22.07	20.31	21.96	19.01	**30.17**
NLSPCA	27.02	27.04	**20.37**	22.10	20.28	21.88	19.00	30.04
NLSPCAbin	27.21	21.10	17.03	21.21	18.45	20.37	18.36	26.96
	Peak = 2							
NLBayes	27.50	24.66	17.13	21.10	19.67	21.34	18.22	21.04
haarTIApprox	27.01	26.43	19.33	23.37	21.72	23.18	19.90	26.53
SAFIR	23.78	26.02	19.25	22.33	21.30	22.74	19.99	28.29
BM3D	28.63	27.70	20.66	24.25	**22.19**	**23.54**	**20.44**	29.75
BM3Dbin	**29.70**	**28.68**	20.01	**24.52**	21.42	23.43	20.17	32.24
NLPCA	29.41	28.02	20.64	23.44	20.75	22.78	19.37	32.25
NLSPCA	29.53	28.11	**20.75**	23.75	20.76	22.86	19.45	**32.35**
NLSPCAbin	27.62	21.13	17.02	21.42	18.33	20.34	18.34	29.31
	Peak = 0.14							
NLBayes	31.17	26.73	22.64	23.61	22.32	23.02	19.60	24.04
haarTIApprox	28.55	28.13	21.16	24.88	22.93	24.23	20.83	28.56
SAFIR	25.40	27.40	20.71	23.76	22.73	23.85	20.88	30.52
BM3D	30.36	29.30	**22.91**	**26.08**	**23.93**	**24.79**	**21.50**	32.50
BM3Dbin	31.15	**30.07**	20.57	25.64	22.00	24.28	20.84	33.52
NLPCA	31.08	29.07	20.96	24.49	20.96	23.18	19.73	**33.73**
NLSPCA	**31.46**	29.51	21.15	24.89	21.08	23.41	20.15	33.69
NLSPCAbin	27.65	21.45	16.00	21.47	18.44	20.35	18.35	29.13



(a) Original (b) Noisy, PSNR=10.00 (c) haarTIApprox, PSNR=24.44 (d) SAFIR, PSNR=24.92 (e) BM3D, PSNR=26.30

(f) AnscombePCA, PSNR=28.29 (g) NLPCA, PSNR=30.75 (h) NLPCAS, PSNR=30.10 (i) BM3Dbin, PSNR=30.45 (j) NLPCASbin, PSNR=28.32

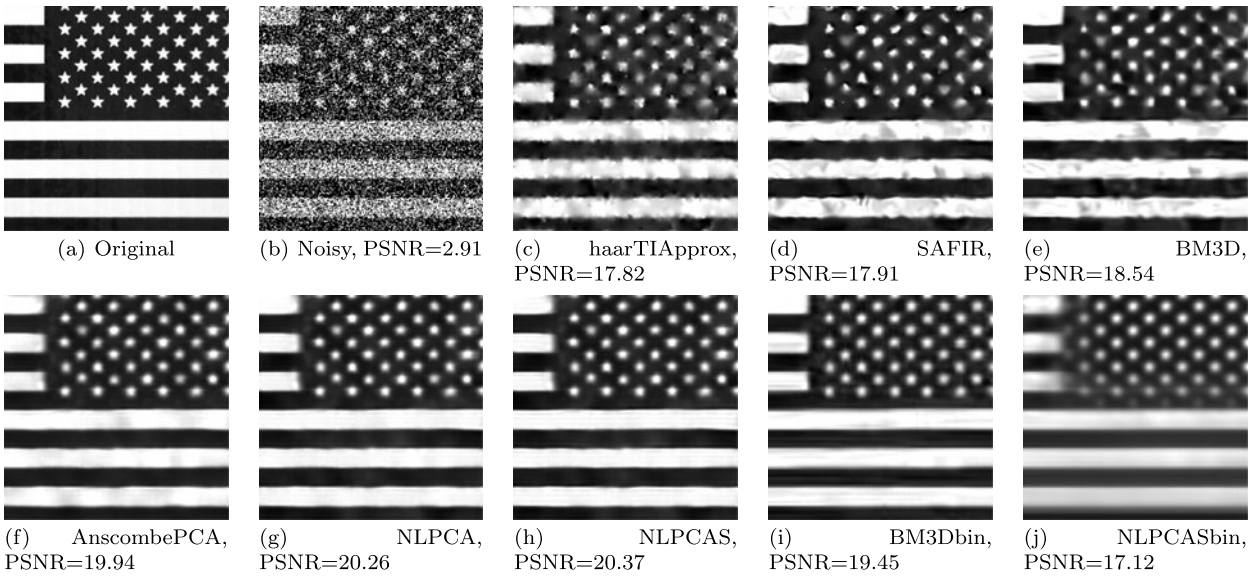
Fig. 5 Toy cartoon image (Ridges) corrupted with Poisson noise with Peak = 1



(a) Original (b) Noisy, PSNR=-7.11 (c) haarTIApprox, PSNR=10.97 (d) SAFIR, PSNR=12.04 (e) BM3D, PSNR=12.92

(f) AnscombePCA, PSNR=13.18 (g) NLPCA, PSNR=14.35 (h) NLPCAS, PSNR=14.40 (i) BM3Dbin, PSNR=13.91 (j) NLPCASbin, PSNR=15.99

Fig. 6 Toy cartoon image (Flag) corrupted with Poisson noise with Peak = 0.1



(a) Original (b) Noisy, PSNR=2.91 (c) haarTIApprox, PSNR=17.82 (d) SAFIR, PSNR=17.91 (e) BM3D, PSNR=18.54

(f) AnscombePCA, PSNR=19.94 (g) NLPCA, PSNR=20.26 (h) NLPCAS, PSNR=20.37 (i) BM3Dbin, PSNR=19.45 (j) NLPCASbin, PSNR=17.12

Fig. 7 Toy cartoon image (Flag) corrupted with Poisson noise with Peak = 1

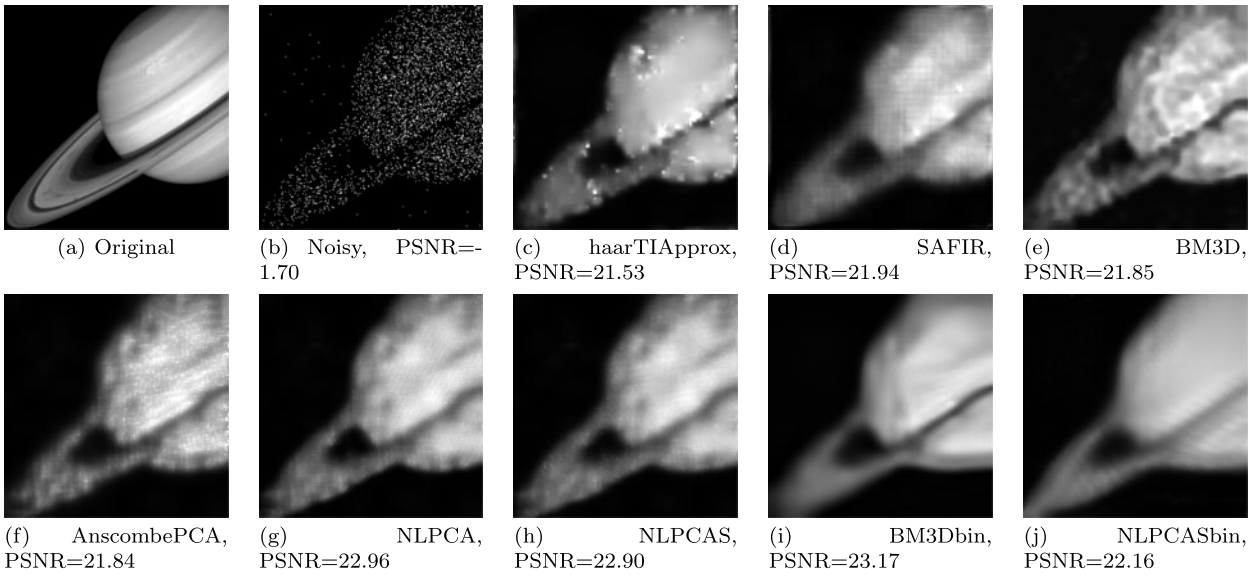
For this image we have also used the 128 first spectral channels, so the data cube is also of size $256 \times 256 \times 128$. Our method removes some of the spurious artifacts generated by the method proposed in [25] and the blurry artifacts in BM4D [28].

7.4 Comparison with Other Methods

7.4.1 Classical PCA with Anscombe Transform

The approximation of the variance provided by the Anscombe transform is reasonably accurate for intensities of three or more (*cf.* Fig. 3 and also [32] Fig. 1-b). In practice this is also the regime where a well-optimized method for Gaussian noise might be applied successfully using this transform and the inverse provided in [31].

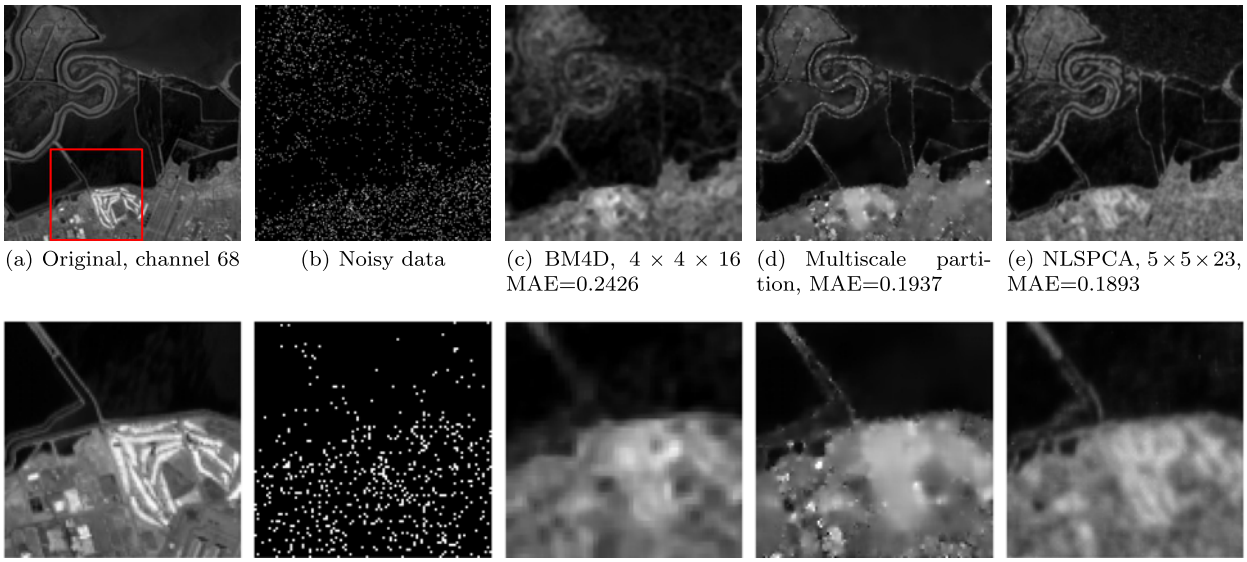
To compare the importance of fully taking advantage of the Poisson model and not using the Anscombe transform, we have derived another algorithm, analogous to our Poisson NLPCA method but using Bregman divergences associated with the natural parameter of a Gaussian random variable instead of Poisson. It corresponds to an implementation similar to the classical power method for computing PCA [10]. The function L to be optimized in (10) is simply re-



(a) Original (b) Noisy, PSNR=-1.70 (c) haarTIApprox, PSNR=21.53 (d) SAFIR, PSNR=21.94 (e) BM3D, PSNR=21.85

(f) AnscombePCA, PSNR=21.84 (g) NLPCA, PSNR=22.96 (h) NLPCAS, PSNR=22.90 (i) BM3Dbin, PSNR=23.17 (j) NLPCASbin, PSNR=22.16

Fig. 8 Toy cartoon image (Saturn) corrupted with Poisson noise with Peak = 0.2



(a) Original, channel 68 (b) Noisy data (c) BM4D, $4 \times 4 \times 16$ MAE=0.2426 (d) Multiscale partition, MAE=0.1937 (e) NLSPCA, $5 \times 5 \times 23$, MAE=0.1893

(f) Original, channel 68 (g) Noisy data (h) BM4D, $4 \times 4 \times 16$, MAE=0.2426 (i) Multiscale partition, MAE=0.1937, (j) NLSPCA, $5 \times 5 \times 23$, MAE=0.1893

Fig. 9 Original and close-up of the red square from spectral band 68 of the Moffett Field. The same methods are considered, and are displayed in the same order: original, noisy (with 0.0387 photons per voxels), BM4D [28] (with inverse Anscombe as in [31]), multiscale partitioning method [25], and our proposed method with patches of size $5 \times 5 \times 23$

placed by the square loss $\tilde{L}$,

$$\tilde{L}(U,V) = \sum_{i=1}^{M}\sum_{j=1}^{N}\big((UV)_{i,j} - Y_{i,j}\big)^2. \tag{22}$$

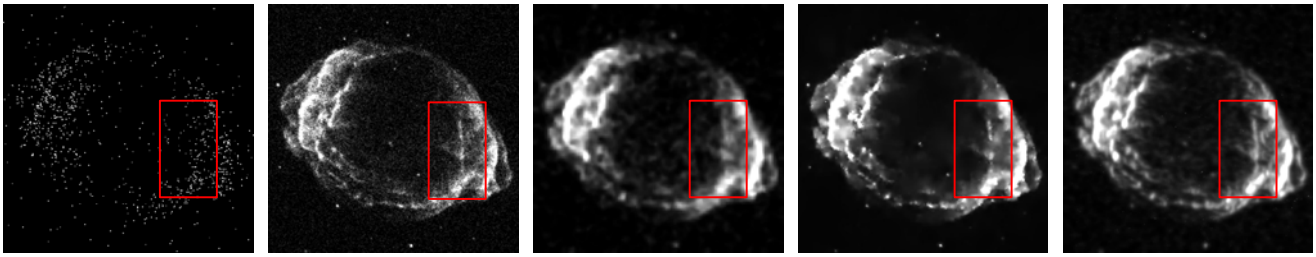
For the Gaussian case, the following update equations are substituted for (13) and (3)

$$U_{t+1,i,:} = U_{t,i,:} - \big((U_tV_t)_{i,:} - Y_{i,:}\big)V_t^\top\big(V_tV_t^\top\big)^{-1}, \tag{23}$$

and

$$V_{t+1,:,j} = V_{t,:,j} - \big(U_{t+1}^\top U_{t+1}\big)^{-1}U_{t+1}^\top\big((U_{t+1}V_t)_{:,j} - Y_{:,j}\big). \tag{24}$$

An illustration of the improvement due to our direct modeling of Poisson noise instead of a simpler Anscombe (Gaussian) NLPCA approach is shown in our previous work [36] and the below simulation results. The gap is most notice-



(a) Noisy (channel 101) (b) Average over channels (c) BM4D, $4 \times 4 \times 16$ (d) Multiscale partition (e) NLSPCA, $5 \times 5 \times 23$

Fig. 10 Spectral image of the supernova remnant G1.9+0.3. We display the spectral band 101 of the noisy observation (with 0.0137 photons per voxels), and this denoised channel with BM4D [28] (with inverse Anscombe as in [31]), the multiscale partitioning method [25], and our proposed method NLSPCA with patches of size $5 \times 5 \times 23$. Note how the highlighted detail shows structure in the average over channels, which appears to be accurately reconstructed by our method

able at low signal-to-noise ratios, and high-frequency artifacts are more likely to appear when using the Anscombe transform. To invert the Anscombe transform we have considered the function provided by [31], and available at http://www.cs.tut.fi/~foi/invansc/. This slightly improves the usual (closed form) inverse transformation, and in our work it is used for all the methods using the Anscombe transform (referred to as Anscombe-NLPCA in our experiments).

7.4.2 Other Methods

We compare our method with other recent algorithms designed for retrieval of Poisson corrupted images. In the case of 2D images we have compared with:

- NLBayes [26] using Anscombe transform and the refined inverse transform proposed in [31].
- SAFIR [5, 22], using Anscombe transform and the refined inverse transform proposed in [31].
- Poisson multiscale partitioning (PMP), introduced by Willett and Nowak [42, 43] using full cycle spinning. We use the haarTIApprox function as available at http://people.ee.duke.edu/~willett.
- BM3D [31] using Anscombe transform with a refined inverse transform. The online code is available at http://www.cs.tut.fi/~foi/invansc/ and we used the default parameters provided by the authors. The version with binning and interpolation relies on 3×3 bins and bilinear interpolation.

In the case of spectral images we have compared our proposed method with

- BM4D [28] using the inverse Anscombe [31] already mentioned. We set the patch size to $4 \times 4 \times 16$, since the patch length has to be dyadic for this algorithm.
- Poisson multiscale partition (PMP for 3D images) [25], adapting the haarTIApprox algorithm to the case of spectral images. As in the reference mentioned, we have considered cycle spinning with 2000 shifts.

For visual inspection of the qualitative performance of each approach, the results are displayed on Fig. 4–10. Quantitative performance in terms of PSNR are given in Table 2.

8 Conclusion and Future Work

Inspired by the methodology of [15] we have adapted a generalization of the PCA [10, 35] for denoising images damaged by Poisson noise. In general, our method finds a good rank-ℓ approximation to each cluster of patches. While this can be done either in the original pixel space or in a logarithmic "natural parameter" space, we choose the logarithmic scale to avoid issues with nonnegativity, facilitating fast algorithms. One might ask whether working on a logarithmic scale impacts the accuracy of this rank-ℓ approximation. Comparing against several state-of-the-art approaches, we see that because our approach often works as well or better than these alternatives, the exponential formulation of PCA does not lose significant approximation power or else it would manifest itself in these results.

Possible improvements include adapting the number of dictionary elements used with respect to the noise level, and proving a theoretical convergence guarantees for the algorithm. The nonconvexity of the objective may only allow convergence to local minima. An open question is whether these local minima have interesting properties. Reducing the computational complexity of NLPCA is a final remaining challenge.

Acknowledgements Joseph Salmon, Zachary Harmany, and Rebecca Willett gratefully acknowledge support from DARPA grant no. FA8650-11-1-7150, AFOSR award no. FA9550-10-1-0390, and NSF award no. CCF-06-43947. The authors would also like to thank J. Boulanger and C. Kervrann for providing their SAFIR algorithm, Steven Reynolds for providing the spectral images from the supernova remnant G1.9+0.3, and an anonymous reviewer for proposing the improvement using the binning step.

Appendix A: Biconvexity of Loss Function

Lemma 1 *The function L is biconvex with respect to (U, V) but not jointly convex.*

Proof The biconvexity argument is straightforward; the partial functions $U \mapsto L(U, V)$ with a fixed V and $V \mapsto L(U, V)$ with a fixed U are both convex. The fact that the problem is non-jointly convex can be seen when U and V are in $\mathbb{R}$ (*i.e.*, $\ell = m = n = 1$), since the Hessian in this case is

$$H_L(U, V) = \begin{pmatrix} V^2 e^{UV} & UVe^{UV} + e^{UV} - Y \\ UVe^{UV} + e^{UV} - Y & U^2 e^{UV} \end{pmatrix}.$$

Thus at the origin one has $H_L(0, 0) = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$, which has a negative eigenvalue, -1. □

Appendix B: Gradient Calculations

We provide below the gradient computation used in Eqs. (13) and (3):

$$\nabla_U L(U, V) = \big(\exp(UV) - Y\big)V^\top,$$
$$\nabla_V L(U, V) = U^\top\big(\exp(UV) - Y\big).$$

Using the component-wise representation this is equivalent to

$$\frac{\partial L(U, V)}{\partial U_{a,b}} = \sum_{j=1}^{N} \exp(UV)_{a,j} V_{b,j} - Y_{a,j} V_{b,j},$$
$$\frac{\partial L(U, V)}{\partial V_{a,b}} = \sum_{i=1}^{M} U_{i,a} \exp(UV)_{i,b} - U_{i,a} Y_{i,b}.$$

Appendix C: Hessian Calculations

The approach proposed by [19, 35] consists in using an iterative algorithm which sequentially updates the jth column of V and the ith row of U. The only problem with this method is numerical: one needs to invert possibly ill conditioned matrices at each step of the loop.

The Hessian matrices of our problems, with respect to U and V respectively are given by

$$\frac{\partial^2 L(U, V)}{\partial U_{a,b} \partial U_{c,d}} = \begin{cases} \sum_{j=1}^{N} \exp(UV)_{a,j} V_{b,j}^2, & \text{if } (a, b) = (c, d), \\ 0 & \text{otherwise,} \end{cases}$$

and

$$\frac{\partial^2 L(U, V)}{\partial V_{a,b} \partial V_{c,d}} = \begin{cases} \sum_{i=1}^{M} U_{i,a}^2 \exp(UV)_{i,b}, & \text{if } (a, b) = (c, d), \\ 0 & \text{otherwise.} \end{cases}$$

Notice that both Hessian matrices are diagonal. So applying the inverse of the Hessian simply consists in inverting the diagonal coefficients.

Appendix D: The Newton Step

In the following we need to introduce the function Vect_C that transforms a matrix into one single column (concatenates the columns), and the function Vect_R that transforms a matrix into a single row (concatenates the rows). This means that

$$\begin{aligned} \mathrm{Vect}_C : \mathbb{R}^{M \times \ell} &\longrightarrow \mathbb{R}^{M\ell \times 1}, \\ U = (U_{1,:}, \ldots, U_{\ell,:}) &\longmapsto \big(U_{1,:}^\top, \ldots, U_{\ell,:}^\top\big)^\top, \end{aligned}$$

and

$$\begin{aligned} \mathrm{Vect}_R : \mathbb{R}^{\ell \times N} &\longrightarrow \mathbb{R}^{1 \times \ell N}, \\ V = \big(V_{:,1}^\top, \ldots, V_{:,\ell}^\top\big)^\top &\longmapsto (V_{:,1}, \ldots, V_{:,\ell}). \end{aligned}$$

Now using the previously introduced notations, the updating steps for U and V can be written

$$\mathrm{Vect}_C(U_{t+1}) = \mathrm{Vect}_C(U_t) - H_{U_t}^{-1} \mathrm{Vect}_C\big(\nabla_U L(U_t, V_t)\big), \tag{25}$$

$$\mathrm{Vect}_R(V_{t+1}) = \mathrm{Vect}_R(V_t) - \mathrm{Vect}_R\big(\nabla_V L(U_t, V_t)\big) H_{V_t}^{-1}. \tag{26}$$

We give the order used to concatenate the coefficients for the Hessian matrix with respect to U, H_U: $(a, b) = (1, 1), \ldots, (M, 1), (1, 2), \ldots (M, 2), \ldots (1, \ell), \ldots, (M, \ell)$. We concatenate the column of U in this order.

It is easy to give the updating rules for the kth column of U, one only needs to multiply the first equation of (25) from the left by the $M \times M\ell$ matrix

$$F_{k,M,\ell,} = \big(0_{M,M}, \ldots, I_{M,M}, \ldots, 0_{M,M}\big) \tag{27}$$

where the identity block matrix is in the kth position. This leads to the following updating rule

$$U_{t+1,\cdot,k} = U_{t,:,k} - D_k^{-1}\big(\exp(U_t V_t) - Y\big)V_{t,k,:}^\top, \tag{28}$$

where D_k is a diagonal matrix of size $M \times M$:

$$D_k = \mathrm{diag}\left(\sum_{j=1}^{n} \exp(U_t V_t)_{1,j} V_{t,k,j}^2, \ldots, \sum_{j=1}^{n} \exp(U_t V_t)_{M,j} V_{t,k,j}^2\right).$$

This leads easily to (13).

By the symmetry of the problem in U and V, one has the following equivalent updating rule for V:

$$V_{t+1,k,:} = V_{t,k,:} - U_{t,:,k}^{\top}\big(\exp(U_t V_t) - Y\big)E_{k,M}^{-1}\,, \tag{29}$$

where E_k is a diagonal matrix of size $N \times N$:

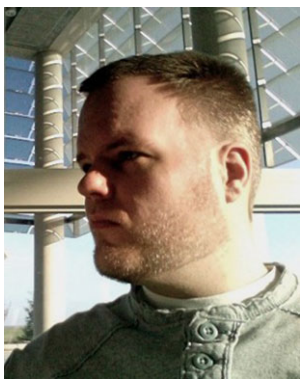
$$E_k = \operatorname{diag}\left(\sum_{i=1}^{M}\exp(U_t V_t)_{i,1}U_{t,i,k}^2, \ldots, \sum_{j=1}^{n}\exp(U_t V_t)_{i,n}U_{t,i,k}^2\right).$$

References

1. Aharon, M., Elad, M., Bruckstein, A.: K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation. IEEE Trans. Signal Process. **54**(11), 4311–4322 (2006)
2. Anscombe, F.J.: The transformation of Poisson, binomial and negative-binomial data. Biometrika **35**, 246–254 (1948)
3. Banerjee, A., Merugu, S., Dhillon, I.S., Ghosh, J.: Clustering with Bregman divergences. J. Mach. Learn. Res. **6**, 1705–1749 (2005)
4. Borkowski, K.J., Reynolds, S.P., Green, D.A., Hwang, U., Petre, R., Krishnamurthy, K., Willett, R.: Radioactive Scandium in the youngest galactic supernova remnant G1.9+0.3. Astrophys. J. Lett. **724**, L161 (2010)
5. Boulanger, J., Kervrann, C., Bouthemy, P., Elbau, P., Sibarita, J.-B., Salamero, J.: Patch-based nonlocal functional for denoising fluorescence microscopy image sequences. IEEE Trans. Med. Imaging **29**(2), 442–454 (2010)
6. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Found. Trends Mach. Learn. **3**(1), 1–122 (2011)
7. Bregman, L.M.: The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. Comput. Math. Math. Phys. **7**(3), 200–217 (1967)
8. Buades, A., Coll, B., Morel, J.-M.: A review of image denoising algorithms, with a new one. Multiscale Model. Simul. **4**(2), 490–530 (2005)
9. Chatterjee, P., Milanfar, P.: Patch-based near-optimal image denoising. In: ICIP (2011)
10. Collins, M., Dasgupta, S., Schapire, R.E.: A generalization of principal components analysis to the exponential family. In: NIPS, pp. 617–624 (2002)
11. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.O.: Image denoising by sparse 3-D transform-domain collaborative filtering. IEEE Trans. Image Process. **16**(8), 2080–2095 (2007)
12. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.O.: BM3D image denoising with shape-adaptive principal component analysis. In: Proc. Workshop on Signal Processing with Adaptive Sparse Structured Representations (SPARS'09) (2009)
13. Danielyan, A., Foi, A., Katkovnik, V., Egiazarian, K.: Denoising of multispectral images via nonlocal groupwise spectrum-PCA. In: CGIV2010/MCS'10, pp. 261–266 (2010)
14. Deledalle, C.-A., Denis, L., Tupin, F.: Poisson NL means: Unsupervised non local means for Poisson noise. In: ICIP, pp. 801–804 (2010)
15. Deledalle, C.-A., Salmon, J., Dalalyan, A.S.: Image denoising with patch based PCA: local versus global. In: BMVC (2011)
16. Figueiredo, M.A.T., Bioucas-Dias, J.M.: Restoration of poissonian images using alternating direction optimization. IEEE Trans. Signal Process. **19**(12), 3133–3145 (2010)
17. Fisz, M.: The limiting distribution of a function of two independent random variables and its statistical application. Colloq. Math. **3**, 138–146 (1955)
18. Fryźlewicz, P., Nason, G.P.: Poisson intensity estimation using wavelets and the Fisz transformation. Technical report, Department of Mathematics, University of Bristol, United Kingdom (2001)
19. Gordon, G.J.: Generalized^2 linear^2 models. In: NIPS, pp. 593–600 (2003)
20. Harmany, Z., Marcia, R., Willett, R.: This is SPIRAL-TAP: sparse poisson intensity reconstruction algorithms—theory and practice. IEEE Trans. Image Process. **21**(3), 1084–1096 (2012)
21. Katkovnik, V., Foi, A., Egiazarian, K.O., Astola, J.T.: From local kernel to nonlocal multiple-model image denoising. Int. J. Comput. Vis. **86**(1), 1–32 (2010)
22. Kervrann, C., Boulanger, J.: Optimal spatial adaptation for patch-based image denoising. IEEE Trans. Image Process. **15**(10), 2866–2878 (2006)
23. Kolaczyk, E.D.: Wavelet shrinkage estimation of certain Poisson intensity signals using corrected thresholds. Stat. Sin. **9**(1), 119–135 (1999)
24. Kolaczyk, E.D., Nowak, R.D.: Multiscale likelihood analysis and complexity penalized estimation. Ann. Stat. **32**(2), 500–527 (2004)
25. Krishnamurthy, K., Raginsky, M., Willett, R.: Multiscale photon-limited spectral image reconstruction. SIAM J. Imaging Sci. **3**(3), 619–645 (2010)
26. Lebrun, M., Colom, M., Buades, A., Morel, J.-M.: Secrets of image denoising cuisine. Acta Numer. **21**(1), 475–576 (2012)
27. Lee, H., Battle, A., Raina, R., Ng, A.Y.: Efficient sparse coding algorithms. In: NIPS, pp. 801–808 (2007)
28. Maggioni, M., Katkovnik, V., Egiazarian, K., Foi, A.: A nonlocal transform-domain filter for volumetric data denoising and reconstruction. IEEE Trans. Image Process. **22**, 119–133 (2013)
29. Mairal, J., Bach, F., Ponce, J., Sapiro, G.: Online learning for matrix factorization and sparse coding. J. Mach. Learn. Res., 19–60 (2010)
30. Mairal, J., Bach, F., Ponce, J., Sapiro, G., Zisserman, A.: Non-local sparse models for image restoration. In: ICCV, pp. 2272–2279 (2009)
31. Mäkitalo, M., Foi, A.: Optimal inversion of the Anscombe transformation in low-count Poisson image denoising. IEEE Trans. Image Process. **20**(1), 99–109 (2011)
32. Mäkitalo, M., Foi, A.: Optimal inversion of the generalized Anscombe transformation for Poisson-Gaussian noise. IEEE Trans. Image Process. **22**, 91–103 (2013)
33. Muresan, D.D., Parks, T.W.: Adaptive principal components and image denoising. In: ICIP, pp. 101–104 (2003)
34. Nielsen, F., Garcia, V.: Statistical exponential families: a digest with flash cards. Arxiv preprint (2009). arXiv:0911.4863
35. Roy, N., Gordon, G.J., Thrun, S.: Finding approximate POMDP solutions through belief compression. J. Artif. Intell. Res. **23**(1), 1–40 (2005)
36. Salmon, J., Deledalle, C.-A., Willett, R., Harmany, Z.: Poisson noise reduction with non-local PCA. In: ICASSP (2012)
37. Salmon, J., Strozecki, Y.: Patch reprojections for non local methods. Signal Process. **92**(2), 447–489 (2012)
38. Salmon, J., Willett, R., Arias-Castro, E.: A two-stage denoising filter: the preprocessed Yaroslavsky filter. In: SSP (2012)
39. Singh, A.P., Gordon, G.J.: Relational learning via collective matrix factorization. In: Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 650–658. ACM, New York (2008)

40. Singh, A.P., Gordon, G.J.: A unified view of matrix factorization models. In: Machine Learning and Knowledge Discovery in Databases, pp. 358–373. Springer, Berlin (2008)
41. Willett, R.: Multiscale analysis of photon-limited astronomical images. In: Statistical Challenges in Modern Astronomy (SCMA) IV (2006)
42. Willett, R., Nowak, R.D.: Platelets: a multiscale approach for recovering edges and surfaces in photon-limited medical imaging. IEEE Trans. Med. Imaging **22**(3), 332–350 (2003)
43. Willett, R., Nowak, R.D.: Fast multiresolution photon-limited image reconstruction. In: Proc. IEEE Int. Sym. Biomedical Imaging—ISBI '04 (2004)
44. Wright, S.J., Nowak, R.D., Figueiredo, M.A.T.: Sparse reconstruction by separable approximation. IEEE Trans. Signal Process. **57**(7), 2479–2493 (2009)
45. Yin, W., Osher, S., Goldfarb, D., Darbon, J.: Bregman iterative algorithms for l1-minimization with applications to compressed sensing. SIAM J. Imaging Sci. **1**(1), 143–168 (2008)
46. Zhang, B., Fadili, J., Starck, J.-L.: Wavelets, ridgelets, and curvelets for Poisson noise removal. IEEE Trans. Image Process. **17**(7), 1093–1108 (2008)
47. Zhang, L., Dong, W., Zhang, D., Shi, G.: Two-stage image denoising by principal component analysis with local pixel grouping. Pattern Recognit. **43**(4), 1531–1549 (2010)
48. Zou, H., Hastie, T., Tibshirani, R.: Sparse principal component analysis. J. Comput. Graph. Stat. **15**(2), 265–286 (2006)

Joseph Salmon received the B.S. in Statistics and Economics from the ENSAE-ParisTech (École Nationale de la Statistique et de l'Administration Économique) and the M.Sc. degree "Random Modelisation" from Univérsité Denis Diderot (P7), Paris, France, 2007. He defended his Ph.D. in 2011 in the Laboratoire de Probabilités et Modéles Aléatoires. He was a Postdoctoral Associsate from 2011 to 2012 in the Nislab at Duke University. Since 2012 he is an assistant professor in machine learning at Telecom ParisTech. His current research interests include statistical aggregation and signal processing.



Zachary Harmany received the B.S. (magna cum lade, with honors) in Electrical Engineering and B.S. (cum lade) in Physics from The Pennsylvania State University in 2006 and Ph.D. in Electrical and Computer Engineering from Duke University in 2012. Since 2012 he has been a Postdoctoral associate at The University of Madison-Wiconsin. His research interests include nonlinear optimization, functional neuroimaging, and signal and image processing with applications in medical imaging, astronomy, and night vision.

Charles-Alban Deledalle received his Engineering degree from Ecole Pour l'Informatique et les Techniques Avancées (EPITA), France, and his M.Sc. degree in science and technology from Université Pierre et Marie Curie (Paris 6), Paris, in 2008. He defended his Ph.D. degree in signal and image processing at Telecom ParisTech, France, in 2011. From 2011 to 2012, he had a postoctoral fellowship at Université Paris Dauphine, France. He is currently a CNRS Researcher Associate at Institut de Matéhmatiques de Bordeaux (IMB), France. His main research interests include image restoration, non-local functionnals and risk estimation. He received the IEEE ICIP Best Student Paper Award in 2010 and the ISIS/EEA/GRETSI Best Ph.D. Award in signal and image processing in 2012.

Rebecca Willett is an associate professor in the Electrical and Computer Engineering Department at Duke University. She completed her Ph.D. in Electrical and Computer Engineering at Rice University in 2005. Prof. Willett received the National Science Foundation CAREER Award in 2007, is a member of the DARPA Computer Science Study Group, and received an Air Force Office of Scientific Research Young Investigator Program award in 2010. Prof. Willett has also held visiting researcher positions at the Institute for Pure and Applied Mathematics at UCLA in 2004, the University of Wisconsin-Madison 2003–2005, the French National Institute for Research in Computer Science and Control (INRIA) in 2003, and the Applied Science Research and Development Laboratory at GE Healthcare in 2002. Her research interests include network and imaging science with applications in medical imaging, wireless sensor networks, astronomy, and social networks. Additional information, including publications and software, are available online.