



HAL
open science

Poisson Shot Noise Removal by an Oracular Non-Local Algorithm

Qiyu Jin, Ion Grama, Quansheng Liu

► **To cite this version:**

Qiyu Jin, Ion Grama, Quansheng Liu. Poisson Shot Noise Removal by an Oracular Non-Local Algorithm. *Journal of Mathematical Imaging and Vision*, 2021, 63 (7), pp.855-874. 10.1007/s10851-021-01033-3 . hal-03430388

HAL Id: hal-03430388

<https://hal.science/hal-03430388>

Submitted on 16 Nov 2021

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

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

Poisson shot noise removal by an oracular non-local algorithm

Qiyu Jin · Ion Grama · Quansheng Liu

Received: date / Accepted: date

Abstract In this paper we address the problem of denoising images obtained under low light conditions for the Poisson shot noise model. Under such conditions the variance stabilization transform (VST) is no longer applicable, so that the state-of-the-art algorithms which are proficient for the additive white Gaussian noise cannot be applied. We first introduce an oracular non-local algorithm and prove its convergence with the optimal rate of convergence under a Hölder regularity assumption for the underlying image, when the search window size is suitably chosen. We also prove that the convergence remains valid when the oracle function is estimated within a prescribed error range. We then define a realisable filter by a statistical estimation of the similarity function which determines the oracle weight. The convergence of the realisable filter is justified by proving that the estimator of the similarity function lies in the prescribed error range with high probability. The experiments show that under low light conditions the proposed filter is competitive compared with the recent state-of-the-art algorithms.

Keywords non-local means · Mean Square Error · Poisson shot noise · oracle estimator

Q. Jin

School of mathematical science, Inner Mongolia University, No. 235 Daxue West Road, Hohhot, 010021, China

E-mail: qyjin2015@aliyun.com

I. Grama

Université de Bretagne-Sud, Campus de Tohaninic, BP 573 56017 Vannes, France

E-mail: ion.grama@univ-ubs.fr

Q. Liu

Université de Bretagne-Sud, Campus de Tohaninic, BP 573 56017 Vannes, France

E-mail: quansheng.liu@univ-ubs.fr

1 Introduction

Noise is inevitable in any image device. A digital imaging system consists of an optical system followed by a photodetector and an associated electronic filter. The photodetector converts the incident optical intensity to a detector current, i.e. photons to electrons. During the process, the true signals are contaminated by many different sources of noise. Poisson shot noise appears in low-light conditions when the number of collected photons is small, such as night vision, medical imaging, underwater imaging, microscopic imaging, optical microscopy imaging, single-particle X-ray free-electron laser (XFEL) diffraction data and astronomy imaging. Such a noise is signal-dependent, and the usual denoising approaches are not adapted.

The key challenge in Poisson intensity estimation problems is that the variances of the observed counts are different. Many authors [2–4, 8, 9, 12, 14, 25, 28, 34, 35, 37] contributed to handle signal-dependent noise, by transforming the image with a Variable-Stabilizing Transformation (VST) such as the Anscombe root transformation [1], the multiscale VSTs [34], the conditional variance stabilization (CVS) [17], or the Fisz transform [13]. All these transformations amount to apply per-pixel non-linearities that effectively reduce the signal dependence of the noise model [27]. Because the transformed signal has an approximate signal independent noise, it may be processed using denoising methods for the Gaussian noise model [7, 22, 36]. After denoising, some inverse transformations, like the Exact Unbiased Inverse (EUI) [24], are applied to the denoised signal, obtaining the estimate of the signal of interest. Rond *et al.* [28] deploys a general approach termed "Plug-and-Play-Prior" to Poisson inverse-problems, which uses an iterative scheme where an easy treatable convex programming algorithm is applied, followed by an efficient Gaussian denoising. Unfortunately, the VST is no longer applicable in low-light conditions, for example when the peak intensity level is less than 5.

There are other approaches which do not use VST. One of them is based on the maximum likelihood principle [6, 16, 30] and does not need the Gaussian approximation. The maximum likelihood estimate of u is obtained by maximizing the log likelihood $\mathcal{P}(u|v)$ of the Poisson probability distribution function (PDF), with respect to $u \geq 0$. Alternatively, one can calculate the maximum likelihood estimate of u by minimizing the negative log likelihood of Poisson PDF given by $\hat{u} = \operatorname{argmin}_{u \geq 0} \{-\ln \mathcal{P}(u|v)\}$. Some methods are modifications of the non-local means filter proposed by Buades *et al.* [5] to denoise images damaged by additive white Gaussian noise. It is based on the similarity phenomenon existing very often in natural images, and assumes that there is sufficient redundant information (pixels having identical noise-free value) in the image to reduce the noise significantly. This filter is known to efficiently reduce the noise and to preserve structures. NLPCA [29] combines Principal Component Analysis (PCA) and sparse Poisson intensity estimation methods in a non-local estimation framework, and SPDA [15] uses dictionary learning within the denoising process. P-LET [23] minimizes the MSE of an unbiased estimate for Poisson shot noise. These approaches are applicable for low peak

level images, but the restoration result is not yet so satisfied. In [20], by minimizing a very tight upper bound of the mean square error (MSE), a non-local means filter with optimal weights, called Optimal Weights Poisson Noise Filter (OWPF), is proposed. This filter was proved to converge at the optimal rate under a suitable regularity condition. It gives a good visual quality of restoration, but it is not robust enough under very low light condition. In fact, with this method the restored images have residual noise when the peak value is very small.

In this paper, we propose a new filter to remove the Poisson shot noise, which is robust under low light condition, and **we prove the convergence** of the proposed filter. The main idea is as follows. We first obtain an oracle estimator by considering a very tight upper bound of the mean square error by varying the size of the search window. Our oracle estimator (see [10] for this concept) depends on the unknown similarity function of the original image. We show that our oracular filter convergences under a Hölder regularity assumption on the underlying image, when the similarity function is estimated within a prescribed error. A calculable algorithm is then constructed by implementing a statistical estimator for the similarity function. The convergence of the obtained filter is ensured by proving that the estimator of the similarity function lies in the prescribed error range with high probability. The experiments show that under low light conditions the new filter is competitive with the recent state-of-the-art algorithms.

Compared with the filter introduced in [20], there are two main differences: (a) in the new filter we use the Gaussian kernel while in [20] we used the triangular kernel; (b) the statistical estimator for the similarity function used in the present paper is also different to that used in [20]. When the peak value is very small (**especially when it is smaller than 1**), the signal is low and the effective number of observations is small. In this case the filter OWPF with optimal weights proposed in [20] is not so efficient: our simulation results show that while applying this filter there is still a lot of residual noise remaining in the restored image, although it keeps well the image details. In this case, with the new proposed filter, the restored image has much less residual noise in the restored image. We think the reason is that the triangular kernel used in OWPF is less smooth than the Gaussian kernel used in the new filter, so that OWPF is less stable with the approximation error of the similarity function, which is important under low light conditions.

It is worth noticing that similar convergence results have been established for Gaussian noise in [19, 21], but with a different similarity function to adapt the Poisson noise, and the proofs are significantly different due to the different nature of noises and similarity functions; see the comments after the statements of Theorems 1, 2 and 3.

The remainder of this paper is organized as following. In Section 2 we introduce an oracle estimator for the Poisson shot noise, and prove its convergence under suitable conditions, with useful information on the rate of convergence. In Section 3 we construct an estimator for the similarity function and estab-

lish for it a convergence theorem. Simulation results are presented with a brief analysis in Section 4.

2 The oracle estimator

2.1 Poisson noise model and notation

For an original digital image u defined on the unit square $\mathbf{I}_0 := [0, 1]^2$, the observed image is a discrete version defined on the lattice $\mathbf{I} := \{0, \frac{1}{N}, \dots, \frac{N-1}{N}\}^2$, with $N \times N$ pixels sampled from \mathbf{I}_0 . The higher the value of N , the higher the resolution of the digital image. The Poisson shot noise model is defined as

$$v(x) \sim \mathcal{P}(u(x)), \quad x \in \mathbf{I}, \quad (1)$$

where v is the observed discrete image defined on the lattice \mathbf{I} . Our objective is to recover the value $u(x_0)$ of the original image for each x_0 in the unit square \mathbf{I}_0 , from the observation v defined on the lattice \mathbf{I} .

For convenience, we extend the definition of the observed image v to the whole unit square \mathbf{I}_0 by setting for all $x = (x_1, x_2) \in \mathbf{I}_0$,

$$v(x) = v(x') \quad \text{with} \quad x' = (x'_1, x'_2) = \left(\frac{[Nx_1]}{N}, \frac{[Nx_2]}{N} \right) \in \mathbf{I}, \quad (2)$$

where $[t]$ denotes the integer part of t . Notice that x' is the closest point in \mathbf{I} of x , which lies in the left and lower side of x . Notice that for all $x \in \mathbf{I}_0$ and $t \in \mathbf{I}$, we have $(x + t)' = x' + t$.

Setting $\varepsilon(x) = v(x) - u(x)$, from the Poisson shot noise model we obtain its additive form

$$v(x) = u(x) + \varepsilon(x), \quad x \in \mathbf{I}, \quad (3)$$

where the noise variable $\varepsilon(x) = v(x) - u(x)$ satisfies

$$\mathbb{E}(\varepsilon(x)) = 0 \quad \text{and} \quad \mathbb{V}ar(\varepsilon(x)) = \mathbb{V}ar(v(x)) = u(x).$$

Compared with the additive Gaussian noise model, the particularity of this additive representation is that the variance of the noise ε is not homogeneous; such a model is sometimes called heteroscedastic.

For any point $x_0 \in \mathbf{I}$ and a positive odd integer d , denote the square window with center x_0 and $d \times d$ pixels of \mathbf{I} by

$$\mathcal{N}_{x,d} = \left\{ y \in \mathbf{I} : \|y - x\|_\infty \leq \frac{d-1}{2N} \right\}, \quad (4)$$

where $\|\cdot\|_\infty$ denotes the supremum norm: $\|z\|_\infty = \max\{|z_1|, |z_2|\}$ for $z = (z_1, z_2)$, and $\frac{d-1}{2N}$ represents half of the edge size of the window. For $x = (x_1, x_2) \in \mathbf{I}_0$, let $x' \in \mathbf{I}$ be as in (2) and define

$$\mathcal{N}_{x,d} = \mathcal{N}_{x',d}.$$

Windows of different sizes will be used. In the following, for $x_0, x \in \mathbf{I}_0$ and positive odd integers d and D , we will use $\mathcal{N}_{x,d}$ for similarity patches and $\mathcal{N}_{x_0,D}$ for search windows. For each $x \in \mathbf{I}_0$, the vector

$$v(\mathcal{N}_{x,d}) = (v(y))_{y \in \mathcal{N}_{x,d}} = (v(x+t))_{t \in \mathcal{N}_{0,d}}$$

formed by the values $v(y)$ of the observed noisy image at pixels $y \in \mathcal{N}_{x,d}$, arranged in the lexicographical order, is called *data patch* or *similarity patch* centered at x . For any $x_0, x \in \mathbf{I}_0$, define

$$\frac{1}{d^2} \|v(\mathcal{N}_{x_0,d}) - v(\mathcal{N}_{x,d})\|_2^2 = \frac{1}{d^2} \sum_{t \in \mathcal{N}_{0,d}} (v(x'_0 + t) - v(x' + t))^2, \quad (5)$$

which measures the similarity between the data patches $v(\mathcal{N}_{x_0,d})$ and $v(\mathcal{N}_{x,d})$.

2.2 The non-local means filter

Let us recall the non-local means filter for Gaussian noise, to well understand its adaptation to the Poisson noise that we will introduce in the following. The non-local means algorithm has been proposed in Buades *et al.* [5] for removing the white Gaussian noise in the additive Gaussian noise model $v(x) = u(x) + \eta(x)$, $x \in \mathbf{I}$, where $u(x)$ is the original image, $v(x)$ is the observed one, $\eta(x)$ is the Gaussian noise which is a sequence of independent random variables of mean $\mathbb{E}(\eta(x)) = 0$ and variance $\mathbb{V}ar(\eta(x)) = \sigma^2$. The parameter $\sigma > 0$ is a measure of the noise level.

For any point $x_0 \in \mathbf{I}_0$, the non-local means filter $\tilde{u}(x_0)$ is defined as weighted means of the values of the observed image in the search window $\mathcal{N}_{x_0,D}$ centered at x_0 :

$$\tilde{u}(x_0) = \sum_{x \in \mathcal{N}_{x_0,D}} w(x_0, x) v(x), \quad (6)$$

where the weight $w(x_0, x)$ is given by

$$w(x_0, x) = e^{-\tilde{\rho}^2(x_0, x)/H^2} / \sum_{y \in \mathcal{N}_{x_0,D}} e^{-\tilde{\rho}^2(x_0, y)/H^2}, \quad (7)$$

with

$$\tilde{\rho}^2(x_0, x) = \sum_{t \in \mathcal{N}_{0,d}} \frac{\kappa(t)(v(x_0+t) - v(x+t))^2}{\sum_{s \in \mathcal{N}_{0,d}} \kappa(s)}.$$

Here $H > 0$ is a bandwidth parameter, and $\kappa(t) > 0$, $t \in \mathcal{N}_{0,d}$ is a fixed kernel function. For $x_0 \in \mathbf{I}_0$ and $x \in \mathbf{I}$, the weight $w(x_0, x)$ measures the similarity of the patches $v(\mathcal{N}_{x_0,d})$ and $v(\mathcal{N}_{x,d})$; more similar the patches, greater the value of the weight.

2.3 Oracle estimator

In order to adapt the non-local means algorithm to the Poisson shot noise, we introduce an oracle estimator (for details on this concept see Donoho and Johnstone (1994 [10]). We define the oracle estimator on \mathbf{I}_0 as follows: for all $x_0 \in \mathbf{I}_0$,

$$u^*(x_0) = u^*(x'_0) = \sum_{x \in \mathcal{N}'_{x'_0, D}} w^*(x'_0, x) v(x), \quad (8)$$

where $x'_0 \in \mathbf{I}$ is defined as in (2), and for $x \in \mathbf{I}$,

$$w^*(x'_0, x) = e^{-\frac{\rho^2(x'_0, x)}{H^2(x'_0)}} \bigg/ \sum_{y \in \mathcal{N}'_{x'_0, D}} e^{-\frac{\rho^2(x'_0, y)}{H^2(x'_0)}}, \quad (9)$$

with

$$\rho(x'_0, x) = |u(x) - u(x'_0)| \quad (10)$$

and $H(x) > 0$ a control function. We set

$$\gamma = \inf\{H(x) : x \in \mathbf{I}_0\}. \quad (11)$$

It is obvious that

$$\sum_{x \in \mathcal{N}'_{x'_0, D}} w^*(x'_0, x) = 1 \quad \text{and} \quad w^*(x'_0, x) \geq 0. \quad (12)$$

The function $\rho(x'_0, x) = |u(x) - u(x'_0)| \geq 0$ measures the similarity of the image brightness between the pixels x and x'_0 ; it is thus called the similarity function. The usual bias-variance decomposition (cf. e.g. [11, 26, 32]) of the Mean Squared Error (MSE) yields

$$\begin{aligned} & \mathbb{E}(u(x_0) - u^*(x_0))^2 \quad (13) \\ &= \left(\sum_{x \in \mathcal{N}'_{x'_0, D}} w^*(x'_0, x) (u(x) - u(x'_0)) \right)^2 + \sum_{x \in \mathcal{N}'_{x'_0, D}} w^*(x'_0, x)^2 u(x) \\ &\leq \left(\sum_{x \in \mathcal{N}'_{x'_0, D}} w^*(x'_0, x) |u(x) - u(x'_0)| \right)^2 + \sum_{x \in \mathcal{N}'_{x'_0, D}} w^*(x'_0, x)^2 u(x). \end{aligned}$$

The inequality (13) can be rewritten in the following form:

$$\mathbb{E}(u(x_0) - u^*(x_0))^2 \leq g(w^*), \quad (14)$$

where

$$g(w^*) = \left(\sum_{x \in \mathcal{N}'_{x'_0, D}} w^*(x'_0, x) \rho(x'_0, x) \right)^2 + \sum_{x \in \mathcal{N}'_{x'_0, D}} w^*(x'_0, x)^2 u(x). \quad (15)$$

We shall define a family of estimates by minimizing the function $g(w^*)$ by changing the width of the search window $\mathcal{N}_{x_0, D}$. Under the Poisson shot noise in low-light conditions, the upper bound of signal function is small, namely the peak intensity

$$\Gamma := \sup\{u(x) : x \in \mathbf{I}_0\} \quad (16)$$

is low.

Definition 1 The image function u on \mathbf{I}_0 is called Hölder if for some constants $L > 0$, $\beta \in (0, 1]$ and all $x, y \in \mathbf{I}_0$,

$$|u(x) - u(y)| \leq L \|x - y\|_\infty^\beta. \quad (17)$$

Let us give some comments on the Hölder assumption. This assumption is rather natural, considering that all optical images are obtained by devices that proceed to a spatial frequency cut-off and are therefore band-limited, hence analytic, so that the Hölder assumption (17) is satisfied with $\beta = 1$. We should mention that in the literature discontinuous image models have also been taken into consideration. This, however, does not diminish the interest in the continuous models. For example the bounded variation (BV) is actually a model of the underlying physical objects emission. It is important to realize that the noise is a discrete perturbation applied to the samples of a smooth function, which may be viewed as the result of the convolution of a BV function with a sinc function. According to Shannon theory, a sampled image is a representation of a continuous image because it is interpolable, by the Shannon-Whittaker formula. A pure BV function like a dark square on white background cannot be sampled as such. It requires previous smoothing.

On the other hand the Hölder assumption of order $\beta \in (0, 1]$ is not so restrictive. For instance, the Brownian motion, which is known as very irregular, satisfies this assumption with $\beta = 1/2$. With $\beta < 1/2$ our model can treat images which are less regular than the Brownian motion.

The following theorem gives the rate of convergence of the oracle estimator with $D \times D$ search windows. Let

$$\Delta = \frac{D-1}{2N} \quad \text{and} \quad n = N^2 \quad (18)$$

be respectively the width (half of the edge size) of the search window and the number of pixels of the image. In the following theorem we prove that when the width of the search window Δ is properly chosen, the Mean Squared Error of the oracle estimator $u^*(x_0)$ converges at the rate $n^{-\frac{\beta}{\beta+1}}$. We will use the notation

$$a_n \asymp b_n$$

to mean that $c_1 b_n \leq a_n \leq c_2 b_n$ for some constants $c_1, c_2 > 0$ and all n .

Theorem 1 *Suppose that the image function u satisfies the local Hölder condition (17) with $\beta \in (0, 1]$. Assume also that $\Delta \asymp n^{-\frac{1}{2\beta+2}}$ and $\gamma \geq cL\Delta^\beta$ for some constant $c > \sqrt{2}$. Then the oracle estimator $u^*(x_0)$ given by (8) satisfies*

$$\mathbb{E}(u^*(x_0) - u(x_0))^2 = O\left(n^{-\frac{\beta}{\beta+1}}\right). \quad (19)$$

For the proof of Theorem 1, see Section 6.1.

The rate of convergence in Theorem 1 is known to be optimal under the stated Hölder condition (see Fan and Gijbels [18]). In view of the definition of Δ (cf. (18)), from Theorem 1 we see that when the search window size D is chosen suitably, the filter converges at optimal rate.

Recall that the parameter γ is a lower bound for the bandwidth $H(x)$ in the oracle estimator (8). The condition $\gamma > cL\Delta^\beta$ that we impose here means that the bandwidth cannot be too small. Such a condition is necessary when we want to prove some rate of convergence of a non-parametric estimator like $u^*(x_0)$. Notice that since Δ converges to 0, the parameter γ can also converge to 0.

Similar results for Gaussian noise can be found in [19, Theorem 3.1] (for the optimal weights with the triangular kernel) and [21, Theorem 2.1] (for the usual weights with the Gaussian kernel). Since the variance of the noise is a constant in the Gaussian case, but varies with the pixel values in the Poisson case considered here, it is more difficult **to deal with Poisson noise rather than with Gaussian noise**.

The next result shows that the rate of convergence of the oracle filter $u^*(x_0)$ remains optimal when $\rho^2(x'_0, x) = |u(x) - u(x'_0)|^2$ is replaced by an approximation $\bar{\rho}^2(x'_0, x)$ with the error

$$e(x'_0, x) := \bar{\rho}^2(x'_0, x) - \rho^2(x'_0, x), \quad x \in \mathcal{N}_{x'_0, D}, \quad x'_0 \in \mathbf{I}_0, \quad (20)$$

satisfying $|e(x'_0, x)| \leq \eta_n$ for some sequence $\eta_n \rightarrow 0$ as $n \rightarrow \infty$.

Theorem 2 *Assume the conditions of Theorem 1. Let $u^*(x_0)$ be given by (8), but with $\rho^2(x'_0, x)$ replaced by an approximation $\bar{\rho}^2(x'_0, x)$, with error satisfying $|\bar{\rho}^2(x'_0, x) - \rho^2(x'_0, x)| \leq Cn^{-\frac{\beta}{\beta+1}}$ for some constant $C > 0$. Then*

$$\mathbb{E}(u^*(x_0) - u(x_0))^2 = O\left(n^{-\frac{\beta}{\beta+1}}\right). \quad (21)$$

For the proof of this theorem see Section 6.2.

A similar result was established in [20, Theorem 2] but for the optimal weights corresponding to the triangular kernel, rather than the Gaussian kernel used in this paper. Another important difference is that the statistical estimator for the similarity function used in the present paper is also different to that in [20]. Accordingly, their proofs are also significantly different. It is worth noting that when we use the Gaussian kernel but with the same estimator of similarity function proposed in [20], then the restoration result is not satisfactory. The proof of Theorem 2 of this paper is significantly different from that of Theorem 2 in [20] because the filter is significantly different, although both theorems give the same optimal rate of convergence.

For Gaussian noise, similar results have been established in [19, Theorem 3.2] (for optimal weights with the triangular kernel) and [21, Theorem 3.1] (for the usual Gaussian weights with the Gaussian kernel). Since Gaussian noise and Poisson noise have very different statistical properties, the approaches are significantly different.

3 Non-local Poisson shot noise filter

3.1 Construction of non-local Poisson shot noise filter

Utilizing the oracle estimator, we construct the non-local Poisson shot noise filter. Let $D > 0$ and $d > 0$. Notice that

$$|u(x) - u(x_0)|^2 = \mathbb{E}|v(x) - v(x_0)|^2 - (u(x_0) + u(x)).$$

By the law of large numbers, a natural estimator of $\mathbb{E}|v(x) - v(x_0)|^2$ is given by $\frac{1}{d^2} \|v(\mathcal{N}_{x_0,d}) - v(\mathcal{N}_{x,d})\|_2^2$ (see (5)), and for each x , $u(x)$ can be estimated by

$$\bar{u}_d(x) := \frac{1}{d^2} \sum_{t \in \mathcal{N}_{0,d}} v(x+t). \quad (22)$$

Therefore, a natural estimator of $|u(x) - u(x_0)|^2$ is

$$\hat{\rho}^2(x_0, x) := \left(\frac{1}{d^2} \|v(\mathcal{N}_{x_0,d}) - v(\mathcal{N}_{x,d})\|_2^2 - \bar{u}_d(x_0) - \bar{u}_d(x) \right)^+. \quad (23)$$

As a result, it is natural to define an adaptive estimator \hat{u} by

$$\hat{u}(x_0) = \sum_{x \in \mathcal{N}_{x_0,D}} \hat{w}(x_0, x) v(x), \quad x_0 \in \mathbf{I}_0, \quad (24)$$

where

$$\hat{w}(x_0, x) = e^{-\frac{\hat{\rho}^2(x_0, x)}{H^2(x_0)}} \Bigg/ \sum_{y \in \mathcal{N}_{x_0,D}} e^{-\frac{\hat{\rho}^2(x_0, y)}{H^2(x_0)}}. \quad (25)$$

The following theorem implies that the similarity function $\rho(x_0, x)$ can be replaced by the estimator $\hat{\rho}(x_0, x)$. Recall that Δ and n are respectively the width of the search window and the number of pixels of the image defined by (18). Let δ be the width of the similarity patches:

$$\delta = \frac{d-1}{2N}. \quad (26)$$

Theorem 3 *Suppose that the function u satisfies the local Hölder condition (17) and $\hat{\rho}^2(x_0, x)$ is given by (23). Assume also that $\Delta \asymp n^{-\frac{1}{2\beta+2}}$ and $\delta \asymp n^{-\alpha}$ for some $\alpha \in (\frac{1-\beta}{2\beta+2}, \frac{1}{2})$. Then there is a constant $c > 0$ such that*

$$\lim_{n \rightarrow \infty} \max_{x_0 \in \mathbf{I}} \max_{x \in \mathcal{N}_{x_0,D}} \mathbb{P} \left\{ |\hat{\rho}^2(x_0, x) - \rho^2(x_0, x)| \leq cn^{-(\frac{1}{2}-\alpha)} \sqrt{\ln n} \right\} = 1. \quad (27)$$

For the proof of this theorem see Section 6.3.

Theorem 3 shows that, when we replace $\rho^2(x_0, x) = |u(x) - u(x_0)|^2$ by its estimator $\hat{\rho}^2(x_0, x)$, the error satisfies

$$|\hat{\rho}^2(x_0, x) - \rho^2(x_0, x)| \leq cn^{-(\frac{1}{2}-\alpha)} \sqrt{\ln n} \leq Cn^{-\frac{\beta}{\beta+1}},$$

with probability close to 1, when $\frac{1-\beta}{2\beta+2} < \alpha < \frac{1}{2}$. Therefore, from Theorems 3 and 2, we see that if we choose suitably the search window size D and similarity patch size d so that $\Delta \asymp n^{-\frac{1}{2\beta+2}}$ and $\delta \asymp n^{-\alpha}$, then the proposed filter \hat{u} converges at the optimal rate $n^{-\frac{\beta}{\beta+1}}$.

A similar result has been established in [21, Theorem 3.2] but for Gaussian noise and for a different similarity function.

4 Simulation results

4.1 Computational algorithm

In this section, we explain how to calculate our filter $\hat{u}(x_0)$, $x_0 \in \mathbf{I}_0$ defined by (24).

First, we calculate the estimator $\hat{\rho}^2(x_0, x)$, $x_0 \in \mathbf{I}_0$ defined by (23). First, to save computation time, we replace the two mean values $\bar{u}(x_0)$, $x_0 \in \mathbf{I}_0$ and $\bar{u}(x)$, $x \in \mathbf{I}$ by the same mean value $\bar{u}_D(x_0)$, $x_0 \in \mathbf{I}_0$ defined by

$$\bar{u}_D(x_0) := \frac{1}{D^2} \sum_{x \in \mathcal{N}'_{x_0, D}} v(x), \quad x_0' \in \mathbf{I}. \quad (28)$$

This avoids the calculation of $\bar{u}(x)$ when x varies; the replacement is reasonable because the averages are close.

Secondly, to better measure the similarity between the patches $v(\mathcal{N}_{x_0, d})$ and $v(\mathcal{N}_{x, d})$, we replace the uniform kernel used in the similarity function of (23) by the non-uniform kernel defined for $t \in \mathcal{N}_{0, d}$ by

$$\kappa(t) = \sum_{k=\max(1, j)}^{\frac{d-1}{2}} \frac{1}{(2k+1)^2}, \quad (29)$$

if $\|t\|_\infty = j$ for some $j \in \{0, 1, \dots, \frac{d-1}{2}\}$.

These considerations lead us to the following version of the estimator (23):

$$\hat{\rho}_\kappa^2(x_0, x) = \left(\sum_{t \in \mathcal{N}_{0, d}} \frac{\kappa(t) |v(x_0 + t) - v(x + t)|^2}{\sum_{t' \in \mathcal{N}_{0, d}} \kappa(t')} - 2\bar{u}_D(x_0) \right)^+. \quad (30)$$

We have therefore the following pseudocode for the calculation of the non-local Poisson shot noise filter (NLPSNF) $\hat{u}(x_0)$, $x_0 \in \mathbf{I}_0$ defined by (24):

Algorithm 1 Non-local Poisson shot noise filter (NLPSNF)**Require:** Noisy image v 1: **Parameters:** D, d, μ, ν .**Ensure:** Estimated image \hat{u} 2: **Step 1**3: **for** each $x_0 \in \mathbf{I}$ **do**4: $\hat{w}(x_0, x) = e^{-\frac{\hat{\rho}_\kappa^2(x_0, x)}{H^2(x_0)}}$, where $H^2(x_0) = \mu\sqrt{u_D(x_0)} + \nu$, $\hat{\rho}_\kappa^2(x_0, x)$ is defined in (30).5: $\hat{u}_1(x_0) = \frac{\sum_{x \in \mathcal{N}_{x_0, D}} \hat{w}(x_0, x)v(x)}{\sum_{x \in \mathcal{N}_{x_0, D}} \hat{w}(x_0, x)}$ 6: **end for**7: **Step 2**8: **for** each $x_0 \in \mathbf{I}$ **do**9: compute $\hat{u}(x_0) = \frac{\sum_{\|x-x_0\| \leq T} e^{-\frac{\|x-x_0\|_2^2}{2}} \hat{u}_1(x)}{\sum_{\|x-x_0\| \leq T} e^{-\frac{\|x-x_0\|_2^2}{2}}}$, where $T = 2$.10: **end for**

Throughout the simulations, this algorithm will be used to compute the non-local means estimator $\hat{u}(x_0)$, $x_0 \in \mathbf{I}_0$.

Finally, we discuss on the choice of the parameters $d, D, H^2(x_0)$. In our simulations, the choice $d = 13$ and $D = 11$ gives good results, for all the tested images. Our experimental results also show that the restoration result is good when the window search size D varies in a suitable range, and that the evolution of the PSNR of the restored image varies as a function of the search window size D in a similar way as in Figure 7 of [21]. In some sense this confirms the conclusion of the convergence theorem which states that when the research window size is chosen properly the filter converges at the optimal rate. We have not tried to find experimentally the exact formula for the optimal choice of D as a function of N and β as the value of β is in general not known.

Sutour *et al.* [31] suggested that a good choice of $H^2(x_0)$ is a linear function of the standard deviation of $\hat{\rho}^2(x_0, x)$. Let us first give an estimate of the variance of $\hat{\rho}^2(x_0, x)$. When the windows $\mathcal{N}_{x_0, d}$ and $\mathcal{N}_{x, d}$ are disjoint, by the independence of $(v(x_0 + t) - v(x + t))^2$, $t \in \mathcal{N}_{0, d}$, we obtain

$$\begin{aligned} \text{Var}(\hat{\rho}^2(x_0, x)) &\approx \text{Var}\left(\frac{1}{d^2} \|v(\mathcal{N}_{x_0, d}) - v(\mathcal{N}_{x, d})\|_2^2 - \bar{u}_d(x_0) - \bar{u}_d(x)\right) \\ &= \frac{1}{d^4} \sum_{t \in \mathcal{N}_{0, d}} \text{Var}(v(x_0 + t) - v(x + t))^2. \end{aligned}$$

Since $v(z)$ is a Poisson variable with mean $u(z)$, for $2 \leq j \leq 4$, we have $\mathbb{E}[v(z)^j] = u(z) + \sum_{2 \leq i \leq j} a_i u(z)^i$ for some absolute constants a_i . Therefore

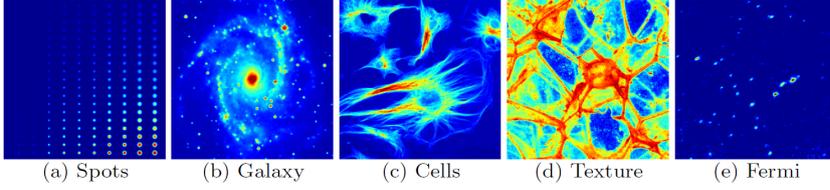


Fig. 1 The five test images used in the experiments.

for some constants c_i , we have

$$\begin{aligned} \text{Var}(\hat{\rho}^2(x_0, x)) &\approx \frac{1}{d^4} \sum_{t \in \mathcal{N}_{0,d}} [u(x_0 + t) + u(x + t)] \\ &\quad + \frac{1}{d^4} \sum_{t \in \mathcal{N}_{0,d}} \sum_{2 \leq i \leq 4} c_i [u^i(x_0 + t) + u^i(x + t)] \\ &= \frac{1}{d^2} [\bar{u}_d(x) + \bar{u}_d(x_0)] + \frac{1}{d^2} \sum_{2 \leq i \leq 4} c_i [\bar{u}_d^i(x) + \bar{u}_d^i(x_0)], \end{aligned}$$

where $\bar{u}_d^i(x) = \frac{1}{d^2} \sum_{t \in \mathcal{N}_{0,d}} u^i(x + t)$ and $\bar{u}_d(x) = \bar{u}_d^1(x)$, for each x . We can normalize the original image so that $u(z) \in [0, 1]$ for all z . Hence for $i \geq 2$, $\bar{u}_d^i(x)$ are negligible with respect to $\bar{u}_d(x)$. Therefore

$$\text{Var}(\hat{\rho}^2(x_0, x)) \approx \frac{1}{d^2} (\bar{u}_d(x) + \bar{u}_d(x_0)) \approx \frac{2\bar{u}_D(x_0)}{d^2}. \quad (31)$$

Therefore, following [31], a good choice of $H^2(x_0)$ is of the form

$$H^2(x_0) = \mu \sqrt{\bar{u}_D(x_0)} + \nu. \quad (32)$$

We add the constant $\nu > 0$ to avoid that $H^2(x_0) = 0$. By doing many simulations, we find that the choice $\mu = 0.2$ and $\nu = 0.0001$ is suitable. This choice can also be justified as follows. We think that the parameter $H^2(x_0)$ should be a function of the standard deviation $\sigma(x_0) = \sqrt{u(x_0)} \approx \sqrt{\bar{u}_D(x_0)}$. With the idea of the affine approximation (Taylor's first order expansion) of a function, it is natural to use an affine relation of type (32), similarly to what we did in our earlier work [21] in removing the Gaussian noise.

Throughout the paper we symmetrize images near the boundary.

4.2 Numerical performance of the non-local Poisson shot noise filter

We have conducted experiments on simulated data on gray scale images which are presented in Fig. 1. Spots, Galaxy and Cells (see Fig. 1 (a)-(c)) were given by Bo Zhang, the author of [34]; Fig. 1 (d) was downloaded from The USC-SIPI Image Database¹. Fig. 1 (e) is a part of Fermi image downloaded from

¹ <http://sipi.usc.edu/database/>

the NASA Fermi support center². Each image is scaled to the peak intensity levels 0.5, 1, 2, 3 and 5, so we focus only at low counts. We summarize our results in the following, both with visual results and performance metrics. We have conducted comparisons of our method and several competing algorithms on simulated data.

As [34], we evaluated the performance of a denoising filter \hat{u} by using the Normalized Mean Integrated Square Error (NMISE) defined by

$$\text{NMISE} = \frac{1}{n^*} \sum_{u(x) > 0, x \in \mathbf{I}} \left(\frac{(\hat{u}(x) - u(x))^2}{u(x)} \right),$$

where $\hat{u}(x)$ are the estimated intensities, $u(x)$ are the respective true values, and $n^* = \text{card} \{u(x) : u(x) > 0, x \in \mathbf{I}\}$. Smaller NMISE values signify better signal restoration.

For each method and each image, we calculate the average and the standard deviation of NMISE over 30 different random noise realizations. Table 1 shows the average NMISE value \pm its standard deviation calculated for the whole images reconstructed by our method and several state-of-the-art approaches, such as P-LET [23], MV+B3 and MV-7/9 [34], P-NLM [9], OWPF [20], E+BM3D [25], NLPKA [29], FoEbin [12] and P⁴IP [28], for the very low light levels of interest. As indicated in the table, our algorithm reaches the best NMISE values in the cases of Cells and Texture with the peak intensity levels 0.5, Fermi with the peak intensity levels 1, 2, 3 and 5; for Spots and Galaxy, when the peak intensity levels are low, our approach is also competitive. The standard deviation values also show that our method is the most stable. In Table 2, we give the rank on the number of the best and second best results in Table 1 about the NMISE comparison of algorithms. From this table we see that our method has the best rank with 13 best results and 8 second best results. Table 3 lists the computation time; it reveals that our code runs fairly quickly. From Tables 2 and 3, we see that our filter and the second best ranked algorithm FoEbin have comparable results, but our filter runs much faster: from Table 3 we can see that the computation time of our filter is 5.6 times shorter than that of FoEbin. The reason is that our filter has no iterative procedure, while FoEbin is an iterative algorithm with high computational complexity.

Visual quality is also important for denoising algorithms. In order to get a better view of the details of restored images, in Figs. 2 - 6, red squares at the end of an arrow of the original and restored images are zoomed twice (the large red squares are the zoomed images). From these figures, we can see that our filter is competitive in details conservation and visual artifacts. More precisely, from Figs. 2 - 4 we see that the restored images by our method conserve better the details and introduce fewer artifacts than other algorithms. In Fig. 5, we see that the method NLPKA has the best visual result, but our method remains competitive. Fig. 6 shows that for the case of Fermi image with $Peak = 5$, our method yields the best visual quality and the lowest NMISE value. In

² <https://fermi.gsfc.nasa.gov/ssc>

Table 1 A comparison of the denoising performance of several denoising algorithms. We give the average NMISE value computed over 30 noise realizations \pm its standard deviation. The best and second best values are indicated as **xxxx** and xxxx respectively.

Algorithm	Peak	Spots	Galaxy	Cells	Texture	Fermi	Average
P-LET[23]	0.5	0.0268 \pm 0.0030	0.0076 \pm 0.0005	0.0140 \pm 0.0004	0.0107 \pm 0.0003	0.0336 \pm 0.0062	0.0185 \pm 0.0021
MV+B3[34]		0.0437 \pm 0.0030	0.0074 \pm 0.0003	0.0188 \pm 0.0007	0.0128 \pm 0.0006	0.0252 \pm 0.0041	0.0216 \pm 0.0017
MV-7/9[34]		0.0330 \pm 0.0021	<u>0.0070</u> \pm 0.0003	0.0125 \pm 0.0007	0.0114 \pm 0.0004	0.0245 \pm 0.0026	0.0177 \pm 0.0012
E+BM3D[25]		0.0441 \pm 0.0038	0.0075 \pm 0.0002	<u>0.0108</u> \pm 0.0004	<u>0.0097</u> \pm 0.0003	0.0221 \pm 0.0022	0.0188 \pm 0.0014
NLPCA[29]		0.0075 \pm 0.0006	0.0069 \pm 0.0008	0.0157 \pm 0.0010	0.0106 \pm 0.0004	0.0211 \pm 0.0032	<u>0.0124</u> \pm 0.0012
FoEbin[12]		31.6557 \pm 0.0000	2.2899 \pm 0.0000	1.3117 \pm 0.0000	0.0397 \pm 0.0000	64.5154 \pm 0.0000	19.9625 \pm 0.0000
P ⁴ IP[28]		0.0277 \pm 0.0042	0.0089 \pm 0.0005	0.0135 \pm 0.0006	0.0104 \pm 0.0003	0.0078 \pm 0.0014	0.0137 \pm 0.0014
OWPF[20]		0.0374 \pm 0.0024	0.0164 \pm 0.0005	0.0187 \pm 0.0006	0.0133 \pm 0.0005	0.0587 \pm 0.0045	0.0289 \pm 0.0017
P-NLM[9]		0.0192 \pm 0.0092	0.0107 \pm 0.0008	0.0123 \pm 0.0006	0.0118 \pm 0.0004	0.0181 \pm 0.0025	0.0144 \pm 0.0027
Ours		<u>0.0149</u> \pm 0.0014	0.0075 \pm 0.0004	0.0105 \pm 0.0004	0.0095 \pm 0.0003	<u>0.0087</u> \pm 0.0033	0.0102 \pm 0.0012
P-LET[23]	1	0.0452 \pm 0.0044	<u>0.0124</u> \pm 0.0004	0.0201 \pm 0.0011	0.0169 \pm 0.0005	0.0457 \pm 0.0037	0.0281 \pm 0.0020
MV+B3[34]		0.0623 \pm 0.0027	0.0127 \pm 0.0004	0.0279 \pm 0.0009	0.0228 \pm 0.0009	0.0427 \pm 0.0029	0.0337 \pm 0.0016
MV-7/9[34]		0.0413 \pm 0.0024	0.0118 \pm 0.0004	0.0188 \pm 0.0006	0.0184 \pm 0.0005	0.0418 \pm 0.0027	0.0264 \pm 0.0013
E+BM3D[25]		0.0563 \pm 0.0049	0.0129 \pm 0.0003	0.0152 \pm 0.0005	<u>0.0149</u> \pm 0.0006	0.0354 \pm 0.0022	0.0269 \pm 0.0017
NLPCA[29]		<u>0.0255</u> \pm 0.0167	0.0127 \pm 0.0009	0.0298 \pm 0.0020	0.0205 \pm 0.0007	0.0348 \pm 0.0033	0.0247 \pm 0.0047
FoEbin[12]		0.0459 \pm 0.0025	0.0101 \pm 0.0003	<u>0.0153</u> \pm 0.0006	0.0146 \pm 0.0004	0.0253 \pm 0.0017	<u>0.0222</u> \pm 0.0011
P ⁴ IP[28]		0.0648 \pm 0.0069	0.0123 \pm 0.0004	0.0199 \pm 0.0008	0.0240 \pm 0.0006	<u>0.0151</u> \pm 0.0016	0.0272 \pm 0.0021
OWPF[20]		0.0305 \pm 0.0013	0.0167 \pm 0.0005	0.0233 \pm 0.0005	0.0175 \pm 0.0004	0.0503 \pm 0.0023	0.0277 \pm 0.0010
P-NLM[9]		0.0347 \pm 0.0114	0.0139 \pm 0.0005	0.0163 \pm 0.0006	0.0161 \pm 0.0005	0.0322 \pm 0.0055	0.0226 \pm 0.0037
Ours		0.0243 \pm 0.0010	0.0128 \pm 0.0004	0.0174 \pm 0.0005	0.0155 \pm 0.0003	0.0133 \pm 0.0019	0.0157 \pm 0.0008
P-LET[23]	2	0.0590 \pm 0.0040	0.0196 \pm 0.0008	0.0359 \pm 0.0124	0.0264 \pm 0.0004	0.0693 \pm 0.0054	0.0420 \pm 0.0046
MV+B3[34]		0.0759 \pm 0.0031	0.0205 \pm 0.0007	0.0413 \pm 0.0016	0.0388 \pm 0.0009	0.0653 \pm 0.0033	0.0484 \pm 0.0019
MV-7/9[34]		0.0480 \pm 0.0022	0.0193 \pm 0.0004	0.0279 \pm 0.0006	0.0288 \pm 0.0007	0.0661 \pm 0.0024	0.0380 \pm 0.0013
E+BM3D[25]		0.0507 \pm 0.0029	0.0169 \pm 0.0004	<u>0.0211</u> \pm 0.0007	<u>0.0240</u> \pm 0.0006	0.0515 \pm 0.0028	0.0328 \pm 0.0015
NLPCA[29]		0.0222 \pm 0.0017	0.0249 \pm 0.0021	0.0581 \pm 0.0041	0.0400 \pm 0.0013	0.0647 \pm 0.0050	0.0420 \pm 0.0028
FoEbin[12]		0.0399 \pm 0.0022	0.0153 \pm 0.0004	0.0210 \pm 0.0007	0.0235 \pm 0.0007	0.0364 \pm 0.0020	<u>0.0272</u> \pm 0.0012
P ⁴ IP[28]		0.1250 \pm 0.0067	0.0223 \pm 0.0006	0.0372 \pm 0.0008	0.0933 \pm 0.0012	<u>0.0329</u> \pm 0.0027	0.0621 \pm 0.0024
OWPF[20]		0.0265 \pm 0.0009	0.0202 \pm 0.0005	0.0320 \pm 0.0006	0.0263 \pm 0.0005	0.0453 \pm 0.0017	0.0301 \pm 0.0008
P-NLM[9]		0.0922 \pm 0.0167	0.0199 \pm 0.0006	0.0234 \pm 0.0007	0.0235 \pm 0.0007	0.0559 \pm 0.0045	0.0430 \pm 0.0046
Ours		<u>0.0257</u> \pm 0.0012	<u>0.0183</u> \pm 0.0004	0.0306 \pm 0.0006	0.0260 \pm 0.0004	0.0315 \pm 0.0012	0.0264 \pm 0.0008
P-LET[23]	3	0.0718 \pm 0.0059	0.0257 \pm 0.0009	0.0730 \pm 0.0814	0.0343 \pm 0.0008	0.0871 \pm 0.0045	0.0584 \pm 0.0187
MV+B3[34]		0.0802 \pm 0.0026	0.0267 \pm 0.0011	0.0511 \pm 0.0010	0.0521 \pm 0.0014	0.0800 \pm 0.0034	0.0580 \pm 0.0019
MV-7/9[34]		0.0525 \pm 0.0019	0.0259 \pm 0.0005	0.0356 \pm 0.0009	0.0372 \pm 0.0007	0.0861 \pm 0.0040	0.0475 \pm 0.0016
E+BM3D[25]		0.0423 \pm 0.0029	<u>0.0222</u> \pm 0.0005	<u>0.0268</u> \pm 0.0006	<u>0.0302</u> \pm 0.0006	0.0622 \pm 0.0033	0.0367 \pm 0.0016
NLPCA[29]		0.0322 \pm 0.0020	0.0382 \pm 0.0047	0.0903 \pm 0.0069	0.0586 \pm 0.0015	0.0943 \pm 0.0048	0.0627 \pm 0.0040
FoEbin[12]		0.0356 \pm 0.0013	0.0204 \pm 0.0004	0.0264 \pm 0.0006	0.0312 \pm 0.0005	<u>0.0440</u> \pm 0.0022	<u>0.0315</u> \pm 0.0010
P ⁴ IP[28]		0.1685 \pm 0.0085	0.0365 \pm 0.0005	0.0608 \pm 0.0009	0.2197 \pm 0.0015	0.0508 \pm 0.0035	0.1073 \pm 0.0030
OWPF[20]		0.0255 \pm 0.0008	0.0241 \pm 0.0006	0.0394 \pm 0.0007	0.0341 \pm 0.0006	0.0470 \pm 0.0015	0.0340 \pm 0.0008
P-NLM[9]		0.0925 \pm 0.0251	0.0255 \pm 0.0008	0.0293 \pm 0.0007	0.0300 \pm 0.0006	0.0739 \pm 0.0063	0.0502 \pm 0.0067
Ours		<u>0.0311</u> \pm 0.0009	0.0204 \pm 0.0004	0.0324 \pm 0.0006	0.0356 \pm 0.0005	0.0335 \pm 0.0013	0.0306 \pm 0.0007
P-LET[23]	4	0.0813 \pm 0.0100	0.0314 \pm 0.0013	0.0612 \pm 0.0248	0.0411 \pm 0.0009	0.1035 \pm 0.0072	0.0637 \pm 0.0088
MV+B3[34]		0.0823 \pm 0.0023	0.0321 \pm 0.0011	0.0603 \pm 0.0011	0.0636 \pm 0.0016	0.0942 \pm 0.0043	0.0665 \pm 0.0021
MV-7/9[34]		0.0563 \pm 0.0022	0.0308 \pm 0.0006	0.0418 \pm 0.0010	0.0443 \pm 0.0008	0.1039 \pm 0.0039	0.0554 \pm 0.0017
E+BM3D[25]		0.0375 \pm 0.0024	<u>0.0260</u> \pm 0.0006	<u>0.0310</u> \pm 0.0007	<u>0.0354</u> \pm 0.0008	0.0715 \pm 0.0033	0.0403 \pm 0.0016
NLPCA[29]		0.0419 \pm 0.0019	0.0510 \pm 0.0049	0.1213 \pm 0.0063	0.0764 \pm 0.0017	0.1266 \pm 0.0148	0.0834 \pm 0.0059
FoEbin[12]		0.0329 \pm 0.0011	0.0238 \pm 0.0006	0.0309 \pm 0.0007	0.0386 \pm 0.0007	0.0513 \pm 0.0029	0.0355 \pm 0.0012
P ⁴ IP[28]		0.2015 \pm 0.0078	0.0548 \pm 0.0009	0.0916 \pm 0.0011	0.3904 \pm 0.0020	0.0689 \pm 0.0035	0.1614 \pm 0.0031
OWPF[20]		0.0257 \pm 0.0008	0.0274 \pm 0.0006	0.0453 \pm 0.0009	0.0408 \pm 0.0006	<u>0.0508</u> \pm 0.0016	<u>0.0380</u> \pm 0.0009
P-NLM[9]		0.0873 \pm 0.0231	0.0301 \pm 0.0010	0.0349 \pm 0.0007	0.0358 \pm 0.0006	0.0906 \pm 0.0063	0.0557 \pm 0.0063
Ours		<u>0.0322</u> \pm 0.0012	0.0262 \pm 0.0005	0.0449 \pm 0.0008	0.0448 \pm 0.0006	0.0452 \pm 0.0015	0.0387 \pm 0.0009
P-LET[23]	5	0.0884 \pm 0.0074	0.0359 \pm 0.0015	0.0700 \pm 0.0189	0.0468 \pm 0.0008	0.1181 \pm 0.0059	0.0718 \pm 0.0069
MV+B3[34]		0.0864 \pm 0.0025	0.0363 \pm 0.0010	0.0686 \pm 0.0012	0.0746 \pm 0.0009	0.1077 \pm 0.0041	0.0747 \pm 0.0019
MV-7/9[34]		0.0596 \pm 0.0022	0.0360 \pm 0.0008	0.0473 \pm 0.0013	0.0504 \pm 0.0008	0.1197 \pm 0.0047	0.0626 \pm 0.0020
E+BM3D[25]		0.0363 \pm 0.0016	0.0297 \pm 0.0007	0.0346 \pm 0.0009	0.0400 \pm 0.0008	0.0799 \pm 0.0036	0.0441 \pm 0.0015
NLPCA[29]		0.0527 \pm 0.0022	0.0620 \pm 0.0063	0.1535 \pm 0.0080	0.0954 \pm 0.0021	0.1518 \pm 0.0072	0.1031 \pm 0.0052
FoEbin[12]		<u>0.0319</u> \pm 0.0009	0.0273 \pm 0.0007	<u>0.0358</u> \pm 0.0006	0.0453 \pm 0.0007	0.0584 \pm 0.0020	0.0397 \pm 0.0010
P ⁴ IP[28]		0.2275 \pm 0.0074	0.0771 \pm 0.0009	0.1292 \pm 0.0010	0.5963 \pm 0.0024	0.0849 \pm 0.0033	0.2230 \pm 0.0030
OWPF[20]		0.0263 \pm 0.0008	0.0307 \pm 0.0006	0.0504 \pm 0.0010	0.0469 \pm 0.0007	<u>0.0572</u> \pm 0.0017	0.0423 \pm 0.0010
P-NLM[9]		0.0844 \pm 0.0109	0.0350 \pm 0.0008	0.0397 \pm 0.0006	<u>0.0412</u> \pm 0.0008	0.1032 \pm 0.0097	0.0607 \pm 0.0046
Ours		0.0338 \pm 0.0011	<u>0.0282</u> \pm 0.0005	0.0411 \pm 0.0007	0.0531 \pm 0.0006	0.0455 \pm 0.0024	<u>0.0403</u> \pm 0.0011

Table 2 The number of the best and second best results in Table 1.

Method	P-LET [23]	MV+B3 [34]	MV-7/9 [34]	E+BM3D [25]	NLPCA [29]	FoEbin [12]	P ⁴ IP [28]	OWPF [20]	P-NLM [9]	Ours
Best	0	0	0	5	3	10	1	3	3	13
2 nd best	1	0	1	10	1	7	2	2	1	8

Table 3 The average computation time in seconds for restoring 256×256 grayscale image. (a) Microsoft Windows 10 Professional (64-bit) (Intel(R) Core(TM) i7-6820HQ CPU @2.70 GHz) with MATLAB 2019b; (b) Microsoft Windows 7 Professional (32-bit) (Intel(R) Core(TM) i3-550 CPU @3.20 GHz) with MATLAB 2015a.

Method	P-LET [23]	MV+B3 [34]	MV-7/9 [34]	E+BM3D [25]	NLPCA [29]	FoEbin [12]	P ⁴ IP [28]	OWPF [20]	P-NLM [9]	Ours
(a)	1.76	–	–	0.89	30.60	35.12	71.08	7.63	–	6.23
(b)	0.95	1.30	2.69	2.59	37.92	–	151.31	15.25	70.71	11.08

Table 4 A PSNR/SSIM value comparison of the denoising performance of several denoising algorithms. The best and second best values are indicated as xxxx and xxxx respectively.

Algorithm	Peak	Spots	Galaxy	Cells	Texture	Fermi	Average
P-LET[23]	4	31.43/0.8230	27.60/0.6663	25.86/0.6258	22.22/0.4410	33.31/0.7834	27.48/0.6579
MV+B3[34]		29.11/0.8236	26.79/0.6735	24.73/0.5740	20.22/0.2787	32.69/0.7614	26.71/0.6223
MV-7/9[34]		30.75/0.8600	27.15/0.6695	26.37/0.6539	21.80/0.3692	33.25/0.7580	27.87/0.6621
E+BM3D[25]		31.04/0.8399	27.43/0.6929	27.10/0.7067	22.64/0.4891	32.32/0.7606	28.11/0.6978
NLPCA[29]		31.49/0.9074	24.79/0.6003	22.29/0.4936	19.54/0.1969	29.86/0.7174	25.60/0.5831
FoEbin[12]		30.90/0.8337	27.89/0.7148	26.70/0.7040	22.36/0.3894	33.30/0.7748	28.23/0.6833
P ⁴ IP[28]		24.26/0.7213	22.53/0.6012	20.03/0.5274	11.54/0.1649	28.49/0.7131	21.37/0.5456
OWPF[20]		31.14/0.8697	26.97/0.6847	24.42/0.6173	21.89/0.3916	33.52/0.7810	27.59/0.6688
P-NLM[9]		30.65/0.8466	27.55/0.6843	26.76/0.6834	22.84/0.4608	33.22/0.7748	28.20/0.6900
Ours		32.08/0.8912	28.21/0.6885	25.61/0.6529	21.93/0.4761	33.35/0.7815	28.24/0.6980
P-LET[23]	5	31.81/0.8369	27.86/0.6558	26.27/0.6425	22.79/0.4822	33.66/0.7908	28.48/0.6816
MV+B3[34]		29.69/0.8436	27.42/0.6904	24.99/0.5912	20.45/0.2975	33.11/0.7764	27.13/0.6398
MV-7/9[34]		31.74/0.8774	27.87/0.6875	26.48/0.6666	22.25/0.4048	33.51/0.7672	28.37/0.6807
E+BM3D[25]		32.12/0.8576	28.04/0.7214	27.80/0.7389	23.20/0.5107	32.52/0.7609	28.75/0.7179
NLPCA[29]		31.25/0.8866	24.94/0.6062	22.17/0.4893	19.67/0.1966	29.90/0.7164	25.59/0.5790
FoEbin[12]		31.35/0.8595	28.46/0.7332	27.30/0.7146	22.64/0.4031	33.65/0.7811	28.68/0.6983
P ⁴ IP[28]		24.15/0.7292	22.04/0.5952	19.43/0.5227	10.71/0.1604	28.66/0.7182	21.00/0.5451
OWPF[20]		31.75/0.8870	27.33/0.7067	25.04/0.6447	22.33/0.4191	33.80/0.7943	28.05/0.6904
P-NLM[9]		31.42/0.8622	27.73/0.6957	27.19/0.6892	23.19/0.4821	33.76/0.7850	28.66/0.7028
Ours		33.78/0.9141	28.55/0.7045	25.81/0.6713	22.05/0.4812	34.12/0.7890	28.86/0.7120

the restored image by our filter, we can see clearly three points in the area of red squares, but these points are seriously blurred or have disappeared in the restored images by other filters.

In the preceding comparisons we have used the NMISE value to measure the quality of restoration. The NMISE value reflects the relative error of restoration, which is well adapted when the peak value is small (less than 5 in our simulations). We have preferred to use this measure rather than the PSNR

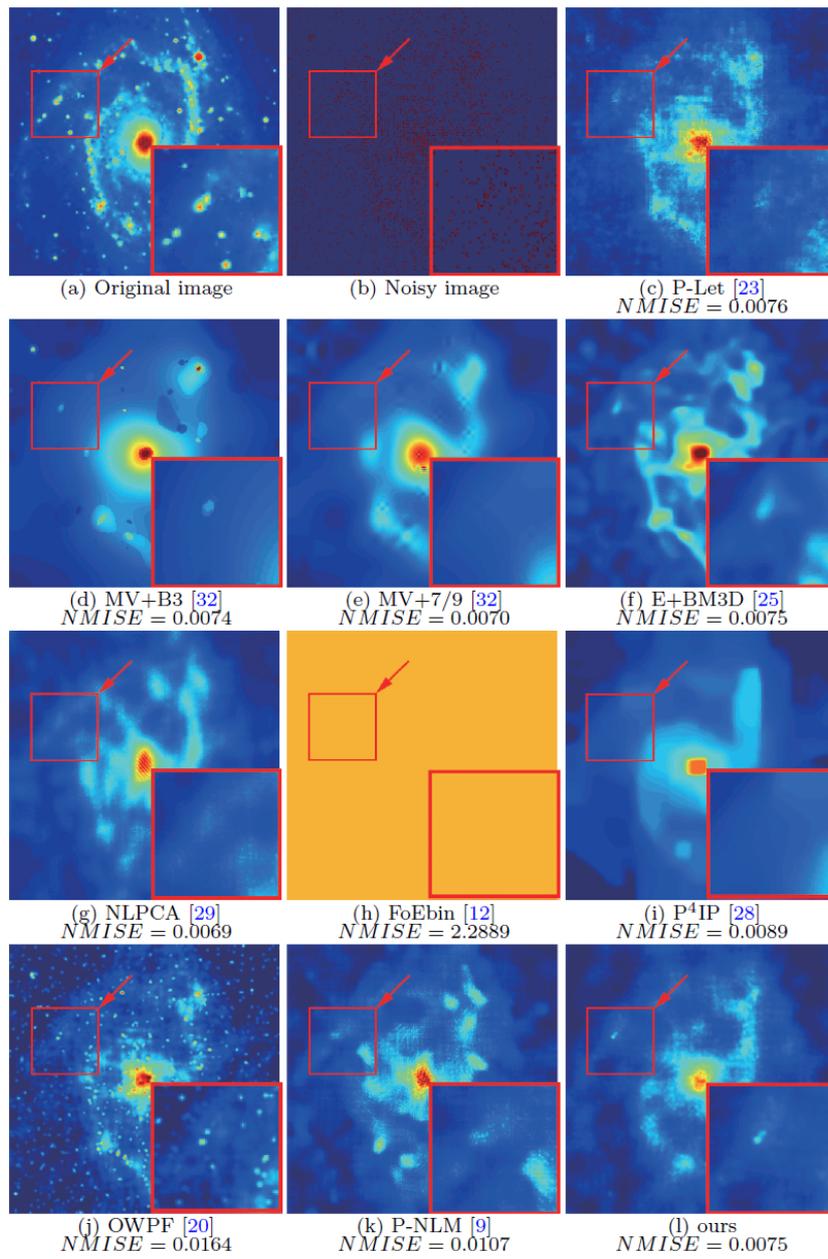


Fig. 2 Denoising of Galaxy with Peak= 0.5, The NMISE value is of the recovered images.

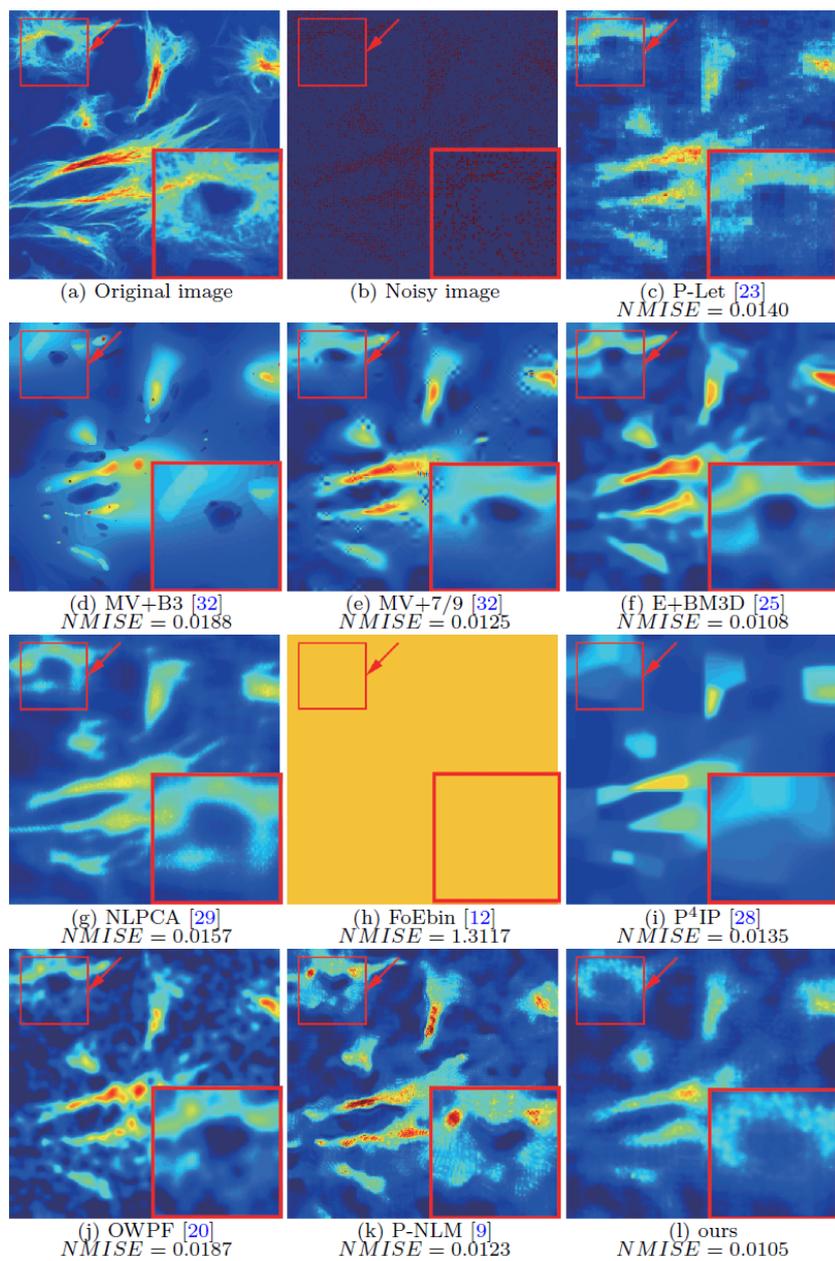


Fig. 3 Denoising of Cells with Peak=0.5, The NMISE value is of the recovered images.

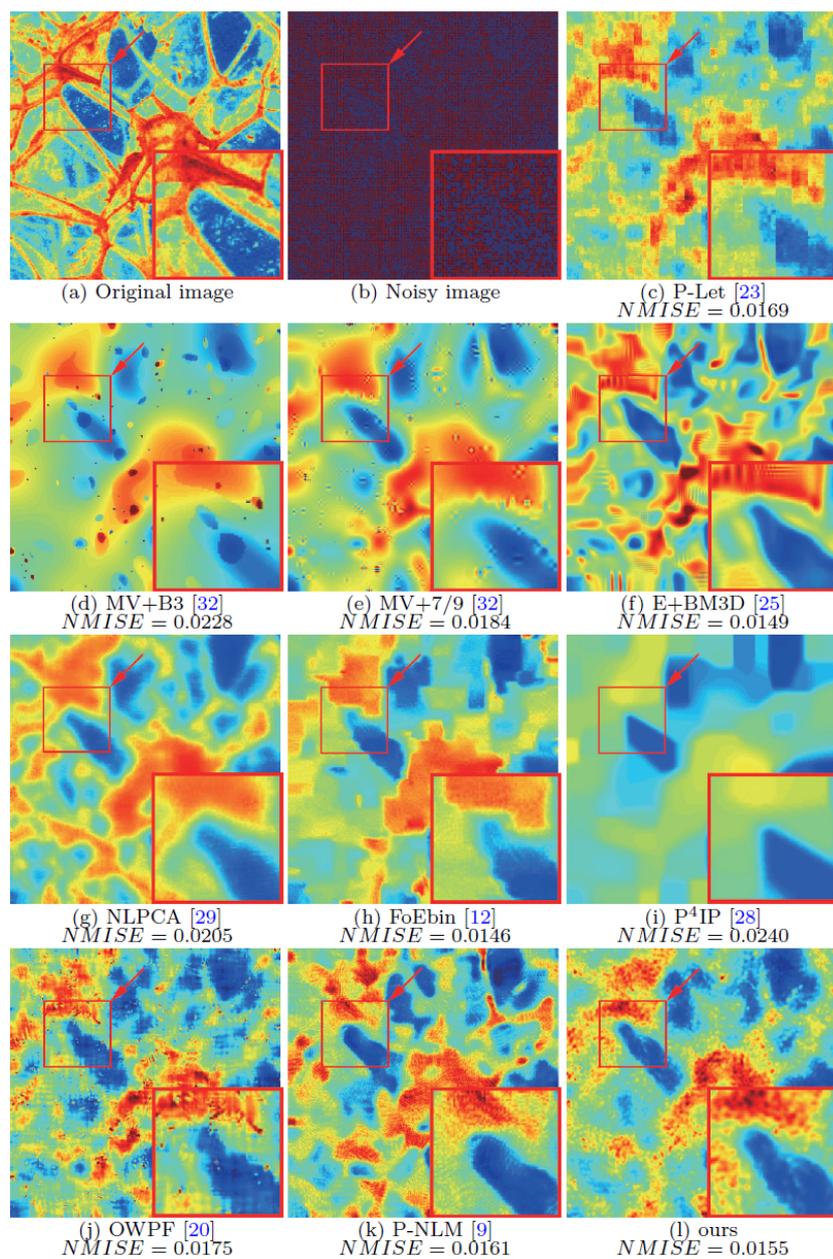


Fig. 4 Denoising of Texture with Peak= 1, The NMISE value is of the recovered images.

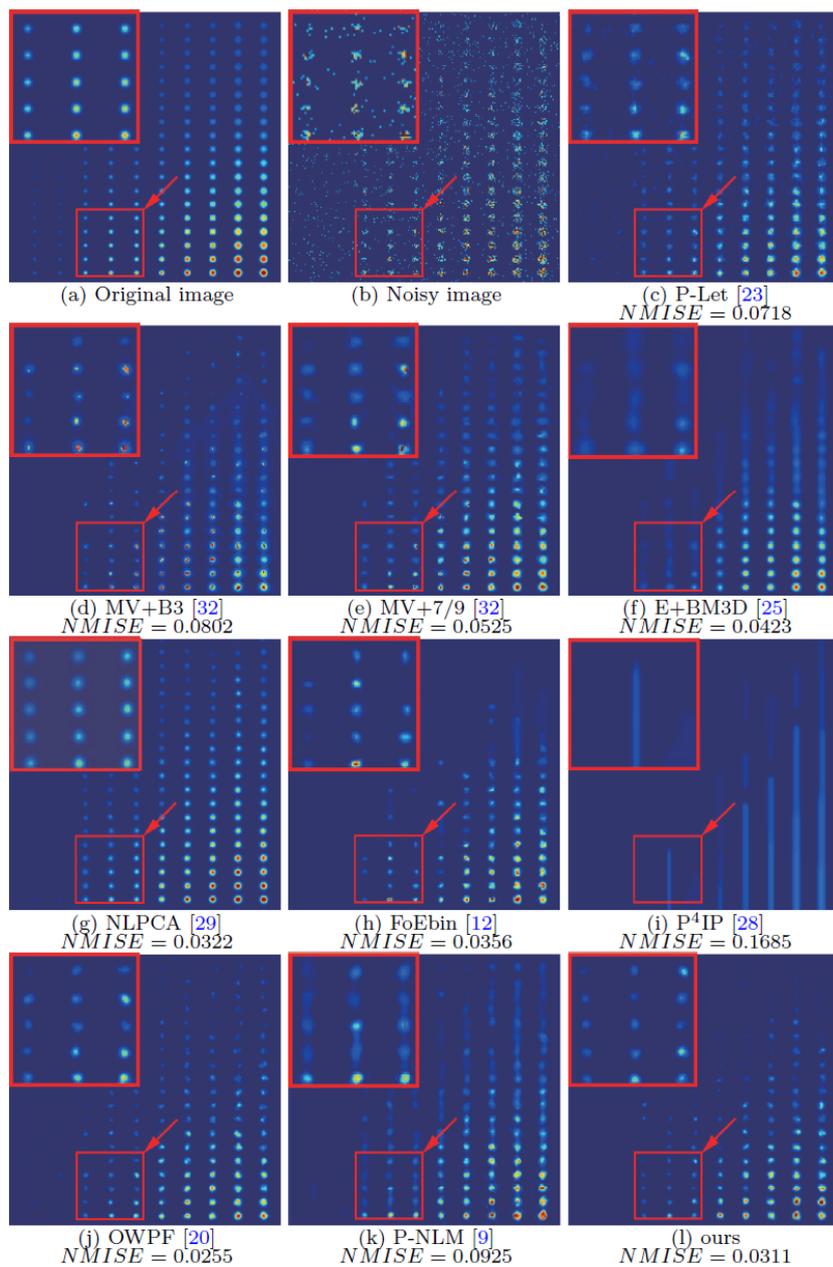


Fig. 5 Denoising of Spots with Peak= 3, The NMISE value is of the recovered images.

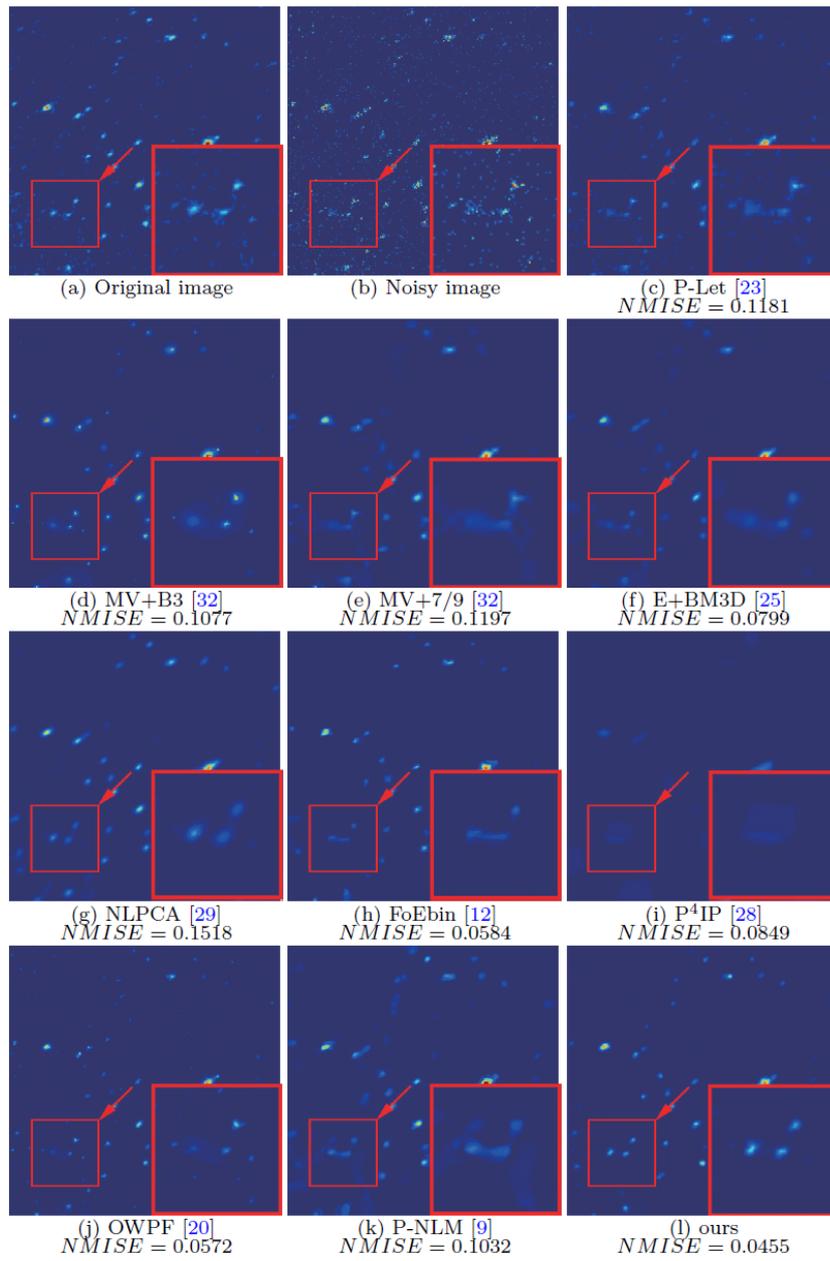


Fig. 6 Denoising of Fermi with Peak=5, The NMISE value is of the recovered images.

(Peak Signal to Noise Ratio) value defined by:

$$\text{PSNR} = 10 \log_{10} \frac{255^2}{\text{MSE}} \quad \text{with} \quad \text{MSE} = \frac{1}{\text{card } \mathbf{I}} \sum_{x \in \mathbf{I}} (\hat{u}(x) - u(x))^2,$$

where u is the original image and \hat{u} the restored one. The reason is that the PSNR value reflects the absolute error of restoration, which is a reliable measure of the restoration quality only when the peak value is sufficiently large (larger than 4). In this case we have calculated the PSNR values, for which the simulation results are coherent to those obtained for the NMISE value: see Table 4, where the SSIM (structural similarity) values defined by Wang *et al.* [33] are also indicated.

The numerical results are performed mainly on simulated noise on a typical type of images, how do they perform on natural images, and real images taken in low light conditions. Also, it may be more helpful to present as well PSNR/SSIM for the quantitatively comparison.

5 Conclusion

In this paper, we present a new non-local image denoising algorithm to deal with the Poisson shot noise model. We first define an oracle filter to remove the Poisson shot noise by an adaptation of the non-local means filter originally introduced for removing the Gaussian noise. We next introduce an estimator of the similarity function and define a calculable filter, called Non-local Poisson shot noise filter, which can effectively remove the Poisson shot noise. We then establish convergence theorems which show that the proposed filter converges at the optimal rate when the search window size is appropriately chosen. Simulation results show that our filter is competitive to remove the Poisson shot noise with low light conditions, and the computation is relatively rapid.

Acknowledgement

The authors are very grateful to Jean-Michel Morel for careful reading, helpful comments and suggestions. They are also grateful to the reviewers for their valuable comments and remarks. The work has been supported by the National Natural Science Foundation of China (Grants Nos. 12061052, 11731012 and 11971063), the Natural Science Fund of Inner Mongolia Autonomous Region (Grant No. 2020MS01002), the China Scholarship Council for a one year visiting at Ecole Normale Supérieure Paris-Saclay (No. 201806810001), the Centre Henri Lebesgue (CHL, ANR-11-LABX-0020-01), and the network information center of Inner Mongolia University. The authors would like to thank Professor Guoqing Chen for providing us some useful suggestions and this work is also partly supported by his project (“111 project” of higher education talent training in Inner Mongolia Autonomous Region).

6 Appendix: Proofs of the main results

6.1 Proof of Theorem 1

We first notice that by the Hölder condition (17), for each $x_0 \in \mathbf{I}_0$, we have $|u(x'_0) - u(x_0)| \leq L\|x'_0 - x_0\|_\infty^\beta \leq LN^{-\beta} = Ln^{-\beta/2}$, where $x'_0 = (x'_{0,1}, x'_{0,2}) = \left(\frac{[Nx_{0,1}]}{N}, \frac{[Nx_{0,2}]}{N}\right) \in \mathbf{I}$. This, together with the elementary inequality $(a+b)^2 \leq 2a^2 + 2b^2$, implies that

$$\begin{aligned} |u^*(x_0) - u(x_0)|^2 &= |u^*(x'_0) - u(x_0)|^2 \\ &\leq 2|u^*(x'_0) - u(x'_0)|^2 + 2|u(x'_0) - u(x_0)|^2 \\ &\leq 2|u^*(x'_0) - u(x'_0)|^2 + 2Ln^{-\beta}. \end{aligned}$$

Since $n^{-\beta} < n^{-\frac{\beta}{\beta+1}}$, it suffices to prove that

$$\mathbb{E}|u^*(x'_0) - u(x'_0)|^2 = O(n^{-\frac{\beta}{\beta+1}}). \quad (33)$$

In other words (since $x'_0 \in \mathbf{I}$), it suffices to prove (19) for each $x_0 \in \mathbf{I}$. So in the following we suppose that $x_0 \in \mathbf{I}$. In this case x'_0 coincides with x_0 .

Denoting for brevity

$$I_1 = \left(\sum_{x \in \mathcal{N}_{x_0, D}} w^*(x_0, x) \rho(x_0, x) \right)^2 = \left(\frac{\sum_{\|x-x_0\|_\infty \leq \Delta} e^{-\frac{\rho^2(x_0, x)}{H^2(x_0)}} \rho(x_0, x)}{\sum_{\|x-x_0\|_\infty \leq \Delta} e^{-\frac{\rho^2(x_0, x)}{H^2(x_0)}}} \right)^2, \quad (34)$$

and

$$\begin{aligned} I_2 &= \sum_{x \in \mathcal{N}_{x_0, D}} (w^*(x_0, x))^2 u(x) \\ &\leq \sum_{x \in \mathcal{N}_{x_0, D}} (w^*(x_0, x))^2 \Gamma \\ &\leq \frac{\Gamma \sum_{\|x-x_0\|_\infty \leq \Delta} e^{-2\frac{\rho^2(x_0, x)}{H^2(x_0)}}}{\left(\sum_{\|x-x_0\|_\infty \leq \Delta} e^{-\frac{\rho^2(x_0, x)}{H^2(x_0)}} \right)^2}, \end{aligned} \quad (35)$$

then we have

$$g(w^*) \leq I_1 + I_2. \quad (36)$$

By the assumption of the theorem $\gamma \geq cL\Delta^\beta (c > \sqrt{2})$, which implies that for $x \in \mathcal{N}_{x_0, D}$, we have

$$\frac{L^2\|x-x_0\|_\infty^{2\beta}}{H^2(x_0)} \leq \frac{L^2\Delta^{2\beta}}{\gamma^2} \leq \frac{1}{c^2}. \quad (37)$$

Noting that $e^{-\frac{\tau^2}{H^2(x_0)}}$, $\tau \in [0, \gamma/\sqrt{2}]$ is decreasing, and using one term Taylor expansion, the inequality (37) implies that

$$\begin{aligned} \sum_{\|x-x_0\|_\infty \leq \Delta} e^{-\frac{\rho^2(x_0, x)}{H^2(x_0)}} &\geq \sum_{\|x-x_0\|_\infty \leq \Delta} e^{-\frac{L^2 \|x-x_0\|_\infty^{2\beta}}{H^2(x_0)}} \\ &\geq \sum_{\|x-x_0\|_\infty \leq \Delta} \left(1 - \frac{L^2 \|x-x_0\|_\infty^{2\beta}}{H^2(x_0)}\right) \\ &\geq D^2 \left(1 - \frac{1}{c^2}\right), \end{aligned} \quad (38)$$

where $D^2 = (2N\Delta + 1)^2$ is the cardinality of the search window (cf. Eq.(18)).

Since $\tau e^{-\frac{\tau^2}{H^2(x_0)}}$ is increasing in $\tau \in [0, \gamma/\sqrt{2}]$,

$$\begin{aligned} \sum_{\|x-x_0\|_\infty \leq \Delta} e^{-\frac{\rho^2(x_0, x)}{H^2(x_0)}} \rho(x_0, x) &\leq \sum_{\|x-x_0\|_\infty \leq \Delta} L \|x-x_0\|_\infty^\beta e^{-\frac{L^2 \|x-x_0\|_\infty^{2\beta}}{H^2(x_0)}} \\ &\leq \sum_{\|x-x_0\|_\infty \leq \Delta} L \|x-x_0\|_\infty^\beta \\ &\leq D^2 L \Delta^2. \end{aligned} \quad (39)$$

The above three inequalities (34), (38) and (39) imply that

$$I_1 \leq \left(\frac{D^2 L \Delta^\beta}{D^2 \left(1 - \frac{1}{c^2}\right)}\right)^2 = c' L^2 \Delta^{2\beta}, \quad \text{where } c' = \left(\frac{c^2}{c^2 - 1}\right)^2. \quad (40)$$

Taking into account (35), (38) and the inequality

$$\sum_{\|x-x_0\|_\infty \leq \Delta} e^{-2\frac{\rho^2(x_0, x)}{H^2(x_0)}} \leq \sum_{\|x-x_0\|_\infty \leq \Delta} 1 = D^2,$$

it is easily seen that

$$I_2 \leq \frac{D^2 \Gamma}{(D^2)^2} \leq \frac{\Gamma}{4\Delta^2 n}. \quad (41)$$

Combining (36), (40), and (41), we get

$$g(w^*) \leq c' L^2 \Delta^{2\beta} + \frac{\Gamma}{4\Delta^2 n}. \quad (42)$$

Using the condition $c_1 n^{-\frac{1}{2\beta+2}} \leq \Delta \leq c_2 n^{-\frac{1}{2\beta+2}}$ for some constants $c_1, c_2 > 0$, from this we infer that

$$g(w^*) \leq \left(c' c_1^{2\beta} L^2 + \frac{\Gamma}{4c_1^2}\right) n^{-\frac{\beta}{\beta+1}}.$$

This ends the proof of (19).

6.2 Proof of Theorem 2

As in the proof of Theorem 1, we can assume that $x_0 \in \mathbf{I}$, so that $x'_0 = x_0$.

By our condition, $e(x, x_0) = \bar{\rho}^2(x_0, x) - \rho^2(x_0, x)$ satisfies $|e(x, x_0)| \leq \eta_n = O(n^{-\frac{\beta}{\beta+1}})$, where $\eta_n = \max_{x_0, x \in \mathbf{I}} |e(x, x_0)|$. Using the elementary inequality $(a - b)^2 \leq |a^2 - b^2|$ for $a, b \geq 0$, we obtain that

$$|\bar{\rho}(x_0, x) - \rho(x_0, x)|^2 \leq |\bar{\rho}^2(x_0, x) - \rho^2(x_0, x)| \leq \eta_n = O(n^{-\frac{\beta}{\beta+1}}).$$

Therefore

$$\bar{\rho}(x_0, x) \leq \rho(x_0, x) + \sqrt{\eta_n}.$$

As $(a + b)^2 \leq 2a^2 + 2b^2$, we have

$$\begin{aligned} \left(\sum_{x \in \mathcal{N}_{x_0, D}} w^*(x_0, x) \bar{\rho}(x_0, x) \right)^2 &\leq \left(\sum_{x \in \mathcal{N}_{x_0, D}} w^*(x_0, x) \rho(x_0, x) + \sqrt{\eta_n} \right)^2 \\ &\leq 2 \left(\sum_{x \in \mathcal{N}_{x_0, D}} w^*(x_0, x) |u(x) - u(x_0)| \right)^2 + 2\eta_n. \end{aligned}$$

Hence

$$\begin{aligned} &\mathbb{E} (u^*(x_0) - u(x_0))^2 \\ &\leq 2 \left(\sum_{x \in \mathcal{N}_{x_0, D}} w^*(x_0, x) |u(x) - u(x_0)| \right)^2 + 2\eta_n + \sum_{x \in \mathcal{N}_{x_0, D}} w^*(x_0, x)^2 u(x). \\ &\leq 2 \left(\left(\sum_{x \in \mathcal{N}_{x_0, D}} w^*(x_0, x) |u(x) - u(x_0)| \right)^2 + \sum_{x \in \mathcal{N}_{x_0, D}} w^*(x_0, x)^2 u(x) \right) + 2\eta_n. \end{aligned}$$

From the proof of Theorem 1 we deduce that

$$\left(\sum_{x \in \mathcal{N}_{x_0, D}} w^*(x_0, x) |u(x) - u(x_0)| \right)^2 + \sum_{x \in \mathcal{N}_{x_0, D}} w^*(x_0, x)^2 u(x) = O\left(n^{-\frac{\beta}{\beta+1}}\right).$$

Since $\eta_n = O(n^{-\frac{\beta}{2+2\beta}})$, we obtain

$$\mathbb{E} (u^*(x_0) - u(x_0))^2 = 2O\left(n^{-\frac{\beta}{\beta+1}}\right) + 2O\left(n^{-\frac{\beta}{\beta+1}}\right) = O\left(n^{-\frac{\beta}{\beta+1}}\right).$$

6.3 Proof of Theorem 3

We first give an expression of $\widehat{\rho}^2(x_0, x)$ defined by (23), which will be suitable for the estimation. For convenience, let

$$A_{x_0, x}(t) = u(x_0 + t) - u(x + t) \quad (43)$$

and

$$\zeta_{x_0, x}(t) = \varepsilon(x_0 + t) - \varepsilon(x + t). \quad (44)$$

With these notations and using (3), we see that the function in the definition of $\widehat{\rho}^2(x_0, x)$ (cf. (23)) can be written as:

$$\begin{aligned} & \|v(\mathcal{N}_{x, d}) - v(\mathcal{N}_{x_0, d})\|_2^2 - \bar{u}(x_0) - \bar{u}(x) \\ &= \frac{1}{d^2} \sum_{t \in \mathcal{N}_{0, d}} (v(x_0 + t) - v(x + t))^2 - \bar{u}(x_0) - \bar{u}(x) \\ &= \frac{1}{d^2} \sum_{y \in \mathcal{N}_{x_0, d}} (u(x_0 + t) - u(x + t) + \varepsilon(x_0 + t) - \varepsilon(x + t))^2 - \bar{u}(x_0) - \bar{u}(x) \\ &= \frac{1}{d^2} \sum_{t \in \mathcal{N}_{0, d}} (A_{x_0, x}(t) + \zeta_{x_0, x}(t))^2 - \bar{u}(x_0) - \bar{u}(x) \\ &= \frac{1}{d^2} \sum_{t \in \mathcal{N}_{0, d}} A_{x_0, x}^2(t) + \frac{1}{d^2} S(x_0, x), \end{aligned}$$

where

$$S(x_0, x) = \sum_{t \in \mathcal{N}_{0, d}} (\zeta_{x_0, x}(t)^2 - u(x_0 + t) - u(x + t) + 2A_{x_0, x}(t)\zeta_{x_0, x}(t)). \quad (45)$$

Therefore, by the definition of $\widehat{\rho}^2(x_0, x)$ (see (23)),

$$\widehat{\rho}^2(x_0, x) = \left(\frac{1}{d^2} \sum_{t \in \mathcal{N}_{0, d}} A_{x_0, x}^2(t) + \frac{1}{d^2} S(x_0, x) \right)^+. \quad (46)$$

We will need two lemmas for the estimation of the two sums in (46).

Lemma 1 *Under the local Hölder condition (17), with Δ and δ defined by (18) and (26), we have*

$$\left| \frac{1}{d^2} \sum_{y \in \mathcal{N}_{0, d}} A_{x_0, x}^2(t) - |u(x) - u(x_0)|^2 \right| \leq 4L^2 \Delta^\beta \delta^\beta.$$

The proof of this lemma can be found in [19, 21].

Lemma 2 *There are two positive constants c_1 and c_2 , depending only on L and Γ , such that for any $0 < z \leq c_1 d$,*

$$\mathbb{P}(|S(x_0, x)| \geq zd) \leq c_2 z^{-2}.$$

Proof Note that the variables

$$X_t = \zeta_{x_0,x}(t)^2 - u(x_0+t) - u(x+t) + 2A_{x_0,x}(t)\zeta_{x_0,x}(t), \quad t \in \mathcal{N}_{0,d} \quad (47)$$

are identically distributed with $\mathbb{E}X_t = 0$. We prove below that the variance $\mathbb{E}X_t^2$ satisfies $\max_{t \in \mathcal{N}_{0,d}} \mathbb{E}X_t^2 \leq b$ for some constant $b > 0$. As $v(x)$ has Poisson law with parameter $u(x)$, it holds that

$$\begin{aligned} \mathbb{E}v(x) &= u(x), \\ \mathbb{E}v^2(x) &= u(x) + u^2(x), \\ \mathbb{E}v^3(x) &= u(x) + 3u^2(x) + u^3(x), \\ \mathbb{E}v^4(x) &= u(x) + 7u^2(x) + 6u^3(x) + u^4(x). \end{aligned}$$

Hence, for each $x \in \mathcal{N}_{x_0,D}$ and each $t \in \mathcal{N}_{0,d}$,

$$\begin{aligned} \mathbb{E}\varepsilon^4(x+t) &= \mathbb{E}(v(x+t) - u(x+t))^4 \\ &= \mathbb{E}v^4(x+t) - 3u(x+t)\mathbb{E}v^3(x+t) + 6u^2(x+t)\mathbb{E}v^2(x+t) \\ &\quad - 3u^3(x+t)\mathbb{E}v(x+t) + u^4(x+t) \\ &= u(x+t) + 4u^2(x+t) + 3u^3(x+t) + 2u^4(x+t) \\ &\leq \Gamma + 4\Gamma^2 + 3\Gamma^3 + 2\Gamma^4, \end{aligned} \quad (48)$$

where the last inequality follows by the definition of Γ (see (16)). From (48) and the inequality $(a+b)^4 \leq 8a^4 + 8b^4$ for $a, b \in \mathbf{R}$, we have

$$\begin{aligned} \mathbb{E}(\zeta_{x_0,x}^4(t)) &= \mathbb{E}(\varepsilon(x_0+t) - \varepsilon(x+t))^4 \\ &\leq \mathbb{E}(8\varepsilon^4(x_0+t) + 8\varepsilon^4(x+t)) \\ &\leq 16(\Gamma + 4\Gamma^2 + 3\Gamma^3 + 2\Gamma^4). \end{aligned} \quad (49)$$

As $\mathbb{E}(\varepsilon(x)) = 0$ and $\text{Var}(\varepsilon(x)) = u(x)$, by the independence of $\varepsilon(x_0+t)$ and $\varepsilon(x+t)$, it follows that

$$\begin{aligned} \mathbb{E}(\zeta_{x_0,x}(t)^2) &= \mathbb{E}(\varepsilon(x_0+t) - \varepsilon(x+t))^2 \\ &= \mathbb{E}\varepsilon^2(x+t) + \mathbb{E}\varepsilon^2(x_0+t) \\ &= u(x+t) + u(x_0+t) \\ &\leq 2\Gamma. \end{aligned} \quad (50)$$

As the function u satisfies the local Hölder condition (17), for $x \in \mathcal{N}_{x_0,x}$

$$A_{x_0,x}^2(t) = (u(x_0+t) - u(x+t))^2 \leq L^2\Delta^{2\beta} \leq L^2. \quad (51)$$

Therefore, taking into account (49), (50) and (51), we obtain, uniformly in $t \in \mathcal{N}_{0,d}$,

$$\begin{aligned} \mathbb{E}X_t^2 &= \mathbb{E}(\zeta_{x_0,x}(t)^2 - u(x_0+t) - u(x+t) + 2A_{x_0,x}(t)\zeta_{x_0,x}(t))^2 \\ &= \mathbb{E}(\zeta_{x_0,x}^4(t)) + (u(x_0+t) + u(x+t))^2 \\ &\quad + 4A_{x_0,x}^2\mathbb{E}(\zeta_{x_0,x}^2(t)) + 2(u(x_0+t) + u(x+t))\mathbb{E}(\zeta_{x_0,x}^2(t)) \\ &\leq 16(\Gamma + 4\Gamma^2 + 3\Gamma^3 + 2\Gamma^4) + 4\Gamma^2 + 4L^2 \times 2\Gamma + 4\Gamma \times \Gamma \\ &= 8(2 + L^2)\Gamma + 72\Gamma^2 + 48\Gamma^3 + 32\Gamma^4. \end{aligned}$$

We have therefore proved that $\mathbb{E}X_t^2 \leq b$, where $b := 8(2 + L^2)\Gamma + 72\Gamma^2 + 48\Gamma^3 + 32\Gamma^4$.

The point in handling the sum $S(x_0, x) = \sum_{t \in \mathcal{N}_{0,d}} X_t$ is that the variables $X_t, t \in \mathcal{N}_{0,d}$ are not necessarily independent. Remark that $\zeta_{x_0,x}(t)$ and $\zeta_{x_0,x}(s)$ are correlated if and only if $t - s = \pm(x_0 - x)$: indeed, it can be easily checked that

$$\mathbb{E}(\zeta_{x_0,x}(t)\zeta_{x_0,x}(s)) = \begin{cases} -\sigma^2, & \text{if } t - s = x_0 - x, \\ \sigma^2, & \text{if } t - s = x - x_0, \\ 0, & \text{otherwise.} \end{cases}$$

By the definition of $\zeta_{x_0,x}(t)$, if $t - s \neq \pm(x - x_0)$, then $\zeta_{x_0,x}(t)$ and $\zeta_{x_0,x}(s)$ are independent, so that X_t and X_s are also independent. Consequently

$$\text{Var}(S(x_0, x)) = \mathbb{E}(S(x_0, x)^2) = \sum_{t,s \in \mathcal{N}_{0,d}} \mathbb{E}(X_t X_s) \quad (52)$$

$$= \sum_{t \in \mathcal{N}_{x_0,d}} \mathbb{E}(X_t^2) + \sum_{t \in \mathcal{N}_{x_0,d}} \sum_{s \in \mathcal{N}_{0,d}: s=t \pm(x-x_0)} \mathbb{E}(X_t X_s). \quad (53)$$

By the Cauchy-Schwarz inequality $\mathbb{E}(X_t X_s) \leq b$. Hence

$$\text{Var}(S(x_0, x)) \leq d^2 b + 2d^2 b = 3d^2 b. \quad (54)$$

Therefore, by Chebyshev's inequality

$$\mathbb{P}(|S(x_0, x)| \geq zd) \leq \frac{\text{Var}(S(x_0, x))}{z^2 d^2} \leq \frac{3b}{z^2}.$$

Now we turn to the proof of Theorem 3. Below c_1, c_2, \dots stand for some constants (independent of n). By equation (26) and the assumption on δ we have $d \geq c_1 n^{\frac{1}{2}-\alpha}$. Applying Lemma 2 with $z = \sqrt{\frac{1}{c_3} \ln n} \leq c_2 d$, we see that

$$\mathbb{P}\left(\frac{1}{d^2} |S(x_0, x)| \geq \frac{\sqrt{\frac{1}{c_3} \ln n}}{d}\right) \leq \frac{c_3}{\ln n}. \quad (55)$$

Therefore,

$$\mathbb{P}\left(\frac{1}{d^2} |S(x_0, x)| \geq c_4 n^{\alpha-\frac{1}{2}} \sqrt{\ln n}\right) \leq \frac{c_3}{\ln n}. \quad (56)$$

By Lemma 1 and the conditions on Δ and δ , we have,

$$\left| \frac{1}{d^2} \sum_{t \in \mathcal{N}_{0,d}} \Lambda_{x_0,x}^2(t) - |u(x) - u(x_0)|^2 \right| \leq 4L^2 \Delta^\beta \delta^\beta \leq c_5 n^{-\frac{\beta}{2\beta+2}-\alpha\beta}. \quad (57)$$

From (46), we see that

$$\hat{\rho}^2(x_0, x) - |u(x) - u(x_0)|^2 \leq \frac{1}{d^2} \sum_{t \in \mathcal{N}_{0,d}} \Lambda_{x_0,x}^2(t) - |u(x) - u(x_0)|^2 + \frac{1}{d^2} |S(x_0, x)|$$

and

$$\widehat{\rho}^2(x_0, x) - |u(x) - u(x_0)|^2 \geq \frac{1}{d^2} \sum_{t \in \mathcal{N}_{0,d}} \Lambda_{x_0,x}^2(t) - |u(x) - u(x_0)|^2 - \frac{1}{d^2} |S(x_0, x)|,$$

so that

$$\left| \widehat{\rho}^2(x_0, x) - |u(x) - u(x_0)|^2 \right| \leq \left| \frac{1}{d^2} \sum_{t \in \mathcal{N}_{0,d}} \Lambda_{x_0,x}^2(t) - |u(x) - u(x_0)|^2 \right| + \frac{1}{d^2} |S(x_0, x)|.$$

Therefore, from (57), we obtain

$$\left| \widehat{\rho}^2(x_0, x) - |u(x) - u(x_0)|^2 \right| \leq c_5 n^{-\frac{\beta}{2\beta+2} - \alpha\beta} + \frac{1}{d^2} |S(x_0, x)|. \quad (58)$$

Combining (56) and (58), we get

$$\mathbb{P} \left(\left| \widehat{\rho}^2(x_0, x) - |u(x) - u(x_0)|^2 \right| \geq c_4 n^{\alpha - \frac{1}{2}} \sqrt{\ln n} + c_5 n^{-\frac{\beta}{2\beta+2} - \alpha\beta} \right) \leq \frac{c_3}{\ln n}.$$

Since the condition $\frac{1}{2(\beta+1)^2} < \alpha < \frac{1}{2}$ implies $\frac{\beta}{2\beta+2} + \alpha\beta > \frac{1}{2} - \alpha > 0$, this implies the inequality (27).

References

1. F. Anscombe. The transformation of Poisson, binomial and negative-binomial data. *Biometrika*, 35(3/4):246–254, 1948.
2. L. Azzari and A. Foi. Variance Stabilization for Noisy+ Estimate Combination in Iterative Poisson Denoising. *IEEE Signal Processing Letters*, 23(8):1086–1090, 2016.
3. L. Azzari and A. Foi. Variance stabilization in Poisson image deblurring. In *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*, pages 728–731. IEEE, 2017.
4. A. Bria, C. Marrocco, L. R. Borges, M. Molinara, A. Marchesi, J.-J. Mordang, N. Karssemeijer, and F. Tortorella. Improving the Automated Detection of Calcifications using Adaptive Variance Stabilization. *IEEE Transactions on Medical Imaging*, 2018.
5. A. Buades, B. Coll, and J. Morel. A review of image denoising algorithms, with a new one. *SIAM Journal on Multiscale Modeling and Simulation*, 4(2):490–530, 2005.
6. E. Chouzenoux, A. Jezierska, J.-C. Pesquet, and H. Talbot. A Convex Approach for Image Restoration with Exact Poisson–Gaussian Likelihood. *SIAM Journal on Imaging Sciences*, 8(4):2662–2682, 2015.
7. K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095, 2007.

8. A. Danielyan, A. Foi, V. Katkovnik, and K. Egiazarian. Denoising of multispectral images via nonlocal groupwise spectrum-PCA. In *Conference on Colour in Graphics, Imaging, and Vision*, number 1, pages 261–266. Society for Imaging Science and Technology, 2010.
9. C.-A. Deledalle, F. Tupin, and L. Denis. Poisson NL means: Unsupervised non local means for Poisson noise. In *2010 IEEE international conference on image processing*, pages 801–804. IEEE, 2010.
10. D. Donoho and J. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425, 1994.
11. J. Fan. Local linear regression smoothers and their minimax efficiencies. *The Annals of Statistics*, pages 196–216, 1993.
12. W. Feng, H. Qiao, and Y. Chen. Poisson noise reduction with higher-order natural image prior model. *SIAM Journal on Imaging Sciences*, 9(3):1502–1524, 2016.
13. M. Fisz. The limiting distribution of a function of two independent random variables and its statistical application. In *Colloquium Mathematicae*, volume 3, pages 138–146. Institute of Mathematics Polish Academy of Sciences, 1955.
14. P. Fryzlewicz. Likelihood ratio Haar variance stabilization and normalization for Poisson and other non-Gaussian noise removal. *arXiv:1701.07263*, 2017.
15. R. Giryes and M. Elad. Sparsity-based Poisson denoising with dictionary learning. *IEEE Transactions on Image Processing*, 23(12):5057–5069, 2014.
16. F. Goudail. Performance comparison of pseudo-inverse and maximum-likelihood estimators of Stokes parameters in the presence of Poisson noise for spherical design-based measurement structures. *Optics letters*, 42(10):1899–1902, 2017.
17. M. Jansen. Multiscale Poisson data smoothing. *J. Roy. Statist. Soc. B*, 68(1):27–48, 2006.
18. F. Jianqing and I. Gijbels. Local polynomial modelling and its applications. *Monographs on Statistics and Applied Probability*. Chapman & Hall/CRC, 1996.
19. Q. Jin, I. Grama, C. Kervrann, and Q. Liu. Nonlocal means and optimal weights for noise removal. *SIAM Journal on Imaging Sciences*, 10(4):1878–1920, 2017.
20. Q. Jin, I. Grama, and Q. Liu. A New Poisson Noise Filter based on Weights Optimization. *Journal of Scientific Computing*, 58(3):548–573, 2014.
21. Q. Jin, I. Grama, and Q. Liu. Convergence theorems for the Non-Local Means filter. *Inverse Problems and Imaging*, 12(4):853–881, 2018.
22. M. Lebrun, A. Buades, and J.-M. Morel. A nonlocal bayesian image denoising algorithm. *SIAM Journal on Imaging Sciences*, 6(3):1665–1688, 2013.
23. F. Luisier, C. Vonesch, T. Blu, and M. Unser. Fast interscale wavelet denoising of Poisson-corrupted images. *Signal Process.*, 90(2):415–427, 2010.

24. M. Makitalo and A. Foi. A closed-form approximation of the exact unbiased inverse of the Anscombe variance-stabilizing transformation. *IEEE Transactions on Image Processing*, PP(99):1–1, 2011.
25. M. Makitalo and A. Foi. Optimal inversion of the Anscombe transformation in low-count Poisson image denoising. *IEEE Transactions on Image Processing*, 20(1):99–109, 2011.
26. J. Mandel. Use of the singular value decomposition in regression analysis. *The American Statistician*, 36(1):15–24, 1982.
27. P. R. Prucnal and B. E. Saleh. Transformation of image-signal-dependent noise into image-signal-independent noise. *Optics letters*, 6(7):316–318, 1981.
28. A. Rond, R. Giryes, and M. Elad. Poisson inverse problems by the plug-and-play scheme. *Journal of Visual Communication and Image Representation*, 41:96–108, 2016.
29. J. Salmon, Z. Harmany, C.-A. Deledalle, and R. Willett. Poisson noise reduction with non-local PCA. *Journal of Mathematical Imaging and Vision*, 48(2):279–294, 2014.
30. R. Srivastava and S. Srivastava. Restoration of Poisson noise corrupted digital images with nonlinear PDE based filters along with the choice of regularization parameter estimation. *Pattern Recognition Letters*, 34(10):1175–1185, 2013.
31. C. Sutour, C.-A. Deledalle, and J.-F. Aujol. Adaptive regularization of the NL-means: Application to image and video denoising. *IEEE Transactions on image processing*, 23(8):3506–3521, 2014.
32. G. R. Terrell and D. W. Scott. Variable kernel density estimation. *The Annals of Statistics*, pages 1236–1265, 1992.
33. Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.
34. B. Zhang, J. Fadili, and J. Starck. Wavelets, ridgelets, and curvelets for Poisson noise removal. *IEEE Transactions on Image Processing*, 17(7):1093–1108, 2008.
35. J. Zhang and K. Hiraoka. Improved Denoising via Poisson Mixture Modeling of Image Sensor Noise. *IEEE Transactions on Image Processing*, PP(99):1–1, 2017.
36. K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017.
37. Y. Zhang, P. Song, and Q. Dai. Fourier ptychographic microscopy using a generalized Anscombe transform approximation of the mixed Poisson-Gaussian likelihood. *Optics express*, 25(1):168–179, 2017.