POISSON NOISE REDUCTION WITH NON-LOCAL PCA

J. Salmon *   *C-A. Deledalle* †   *R. Willett* *   *Z. Harmany**

* Duke University, ECE Department
Durham, NC, USA
†CEREMADE, CNRS-Paris-Dauphine,
Paris, France

ABSTRACT

Photon limitations arise in spectral imaging, nuclear medicine, astronomy and night vision. The Poisson distribution used to model this noise has variance equal to its mean so blind application of standard noise removals methods yields significant artifacts. Recently, overcomplete dictionaries combined with sparse learning techniques have become extremely popular in image reconstruction. The aim of the present work is to demonstrate that for the task of image denoising, nearly state-of-the-art results can be achieved using small dictionaries only, provided that they are learned directly from the noisy image. To this end, we introduce patch-based denoising algorithms which perform an adaptation of PCA (Principal Component Analysis) for Poisson noise. We carry out a comprehensive empirical evaluation of the performance of our algorithms in terms of accuracy when the photon count is really low. The results reveal that, despite its simplicity, PCA-flavored denoising appears to be competitive with other state-of-the-art denoising algorithms.

Index Terms— Image denoising, gradient methods, Newton's method, signal representations.

1. INTRODUCTION, MODEL AND NOTATION

Since the introduction of patch based method for image denoising [1], those methods have proved to outperform previously considered approaches [2, 3]. Our work is inspired by recent combining Principal Component Analysis (PCA) with patch based approaches [4, 5, 6] for the Additive White Gaussian Noise (AWGN) model.

One important difference with the standard AWGN model is that the effect of Poisson noise increases (i.e. the signal-to-noise ratio decreases) as the mean intensity value decreases. However, most state-of-the-art methods dealing with Poisson noise rely on variance stabilization techniques as in [7] and [8]. Those two methods consist in using the Anscombe transform [9] and to treat the processed image as if it was corrupted by a Gaussian noise. A recent modification of the inverse Anscombe transform [10] has shown impressive results on standard images.

Our main contribution is to combine Poisson-PCA (also referred to as Exponential-PCA) [11, 12] with patches to denoise images corrupted by Poisson noise. We will detail the targeted optimization problem and present results improving state-of-the-art methods when the noise level is particularly high, a range of noise where the usual treatment relying on Anscombe's [9] transform is no longer relevant. We coined our method Poisson NL-PCA, (for Non-Local Principal Component Analysi).

A major difference with our method is that we directly handle the Poisson structure of the noise, without any "Gaussianization" of the data, a point of view also adopted in [13]. These direct approach is particularly relevant when the image suffers from a high noise level (i.e., low photon emission). We apply patch based methods (also referred to as Non-Local methods) to Poisson noise, adapting PCA-based denoising in this heteroscedastic context.

Several methods have been proposed to represent the data in a lower dimensional space in the same spirit as PCA. The framework considered in [11, 12] deals with data well approximated by random variables drawn from any exponential distribution. A simple case is the standard PCA, assuming the random variables are distributed according to a Gaussian distribution.

1.1. Problem formulation

For $i = 1, \ldots, M$, let y_i be the observed pixel values obtained through an image acquisition device. We consider each x_i to be an independent random Poisson variable whose mean $x_i \geq 0$ is the underlying intensity value to be estimated. Explicitly, the discrete Poisson probability of each y_i is

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

Recall the objective of Poisson-PCA. For that, we denote by Y the $M \times N$ matrix of all the (vectorized) $\sqrt{N} \times \sqrt{N}$ overlapping patches extracted from the noisy image and by $[\![1, M]\!]$ the set $\{1, \ldots, M\}$. Then, one aims to approximate Y by :

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

where

- U is the $M \times \ell$ matrix of coefficients.
- V is the $\ell \times N$ matrix representing the dictionary components / axis. The rows of V represents the dictionary elements.
- $\exp(UV)$ is the element-wise exponential transform of UV:

$$\forall (i,j) \in [\![1, M]\!] \times [\![1, N]\!], \, \exp(UV)_{i,j} = \exp\left([UV]_{i,j}\right). \tag{3}$$

We assume that the patches, up to an exponential transform, can be approximated by a low dimensional space of small dimension ℓ, i.e., we assume that ℓ is small with respect to M.

The framework introduced in [11, 12] leads in the Poisson case to minimizing the following loss function L:

$$L(U, V) = \sum_{i=1}^{M} \sum_{j=1}^{N} \exp(UV)_{i,j} - Y_{i,j}(UV)_{i,j} \quad , \tag{4}$$

with respect to the matrices U and V. Defining the corresponding minimizers of the non-jointly convex problem

$$(U^*, V^*) = \underset{(U,V) \in \mathbb{R}^{M \times \ell} \times \mathbb{R}^{\ell \times N}}{\arg\min} L(U, V), \tag{5}$$

the original data is then denoised by considering $\hat{Y} = \exp(U^*V^*)$.

We restrict the dictionary elements to be in the Euclidean unit sphere since otherwise the solution of the problem is obviously non unique, and multiplying the coefficients by some positive number and dividing the dictionary by the same number does not modify L: $L(U, V) = L(tU, \frac{V}{t})$ for any $t \neq 0$.

2. NEWTON'S METHOD FOR MINIMIZING L

Here we follow the approach proposed by [14, 15] that consists in using Newton steps to minimize the function L. Though the joint function L is not convex, when fixing one variable and keeping the other fixed the partial optimization problem is convex. Therefore we consider Newton updates on the partial problems. It consists in applying a gradient descent step where the updates matrix used are the inverse of the Hessian matrices. Thus, to apply the Newton's method, one needs to invert the Hessian matrices, with respect to both variable 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

$$[H_U]_{(a,b),(c,d)} = \begin{cases} \sum_{j=1}^{N} \exp(UV)_{a,j} V_{b,j}^2, & \text{if } (a,b) = (c,d), \\ 0 & \text{otherwise.} \end{cases}$$
(6)

$$[H_V]_{(a,b),(c,d)} = \begin{cases} \sum_{i=1}^{M} U_{i,a}^2 \exp(UV)_{i,b}, & \text{if } (a,b) = (c,d), \\ 0 & \text{otherwise.} \end{cases}$$
(7)

where both matrices are diagonal, and so are their inverses.

We propose to update the rows of U and columns of V as proposed in [15]. 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). The updating step for U and V are then

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

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

Easy algebra (cf. [14] for more details) leads to the following updating rules for $U_{t+1,i,:}$ the ith row of U_{t+1}:

$$U_{t+1,i,:} = U_{t,i,:} - (\exp(U_t V_t)_{i,:} - Y_{i,:}) V_t^\top (V_t D_i V_t^\top)^{-1}, \quad (8)$$

where $D_i = \text{diag}\left(\exp(U_t V_t)_{i,1}, \ldots, \exp(U_t V_t)_{i,N}\right)$ 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

$$V_{t+1,:,j} = V_{t,:,j} - (U_{t+1}^\top E_j U_{t+1})^{-1} U_{t+1}^\top (\exp(U_{t+1} V_t)_{:,j} - Y_{:,j}),$$
(9)

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

In practice we have iterated this alternating process until reaching $\| \exp(U_t V_t) - \exp(U_{t+1} V_{t+1}) \|^2 / \| \exp(U_t V_t) \|^2 \leq \varepsilon_{\text{stop}}$ for some (small) real number $\varepsilon_{\text{stop}}$. Moreover for numerical stability we have added a Tikhonov/Ridge regularization term, thus we substitute $(V_t D_i V_t^\top + \varepsilon_{\text{cond}} I_\ell)$ to $V_t D_i V_t^\top$ in Eq. (8), respectively $(U_t^\top E_j U_t) + \varepsilon_{\text{cond}} I_\ell)$ to $(U_t^\top E_j U_t)$ in Eq. (9).

Algorithm 1 Poisson NL-PCA

Inputs: noisy image I
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 $\hat{I}$
Patchization: create the collection of patches for the noisy image
Clusterization: create K clusters of patches using K-Means
The k-th 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 (8)
        end for
        for all $j \leq \ell$ **do**
            Update the jth column of V using (9)
        end for
        $t := t + 1$
    end while
    $\hat{Y}^k = \exp(U_t V_t)$
end for
Concatenation: get the whole collection of denoised patches $\hat{Y}$
Reprojection: average the various pixel estimates due to overlaps to get an image estimate: $\hat{I}^{\text{int}}$
Second iteration: repeat the process, using the denoised image to clusterize the noisy patches: $\hat{I}$

---

3. CLUSTERING AND DOUBLE ITERATION

One strategy could be to use patches from the whole image, but a more robust approach consists by first performing a clustering step. It avoids grouping dissimilar regions in the image. Enforcing similarity insides those groups allows us to use a lower rank representation of the data.

For this step, we have simply used the K-means algorithm, with a small number K of clusters. More refined methods, adapted to the structure of the Poisson noise, could also be used for this step.

In the high noise setting we are targeting, the noise might be so strong that the clustering of the patches on the noisy observations is not precise enough. We have thus chosen to perform a second step, so that the noisy patches are clustered with respect to the similarity measured on their denoised versions. This first pre-filtering can be done using any method, but for the sake of clarity we have applied the same method in those two steps. In practice, we have notice an improvement of the performance from 0.5 dB up to 1.5 dB in term of PSNR, depending on the noise level (see Tab. 1).

4. EXPERIMENTAL SETTINGS

Here we present a comparison of our methods with other state-of-the-art approaches. All our results use the following parameters for our Poisson NL-PCAalgorithm :

- N = 20×20 : dimension of the (vectorized) patches
- ℓ = 4   : number of components kept
- K = 14   : number of clusters
- N = 10^{-1} : stopping criterion in Newton's method
- N_{iter} = 20   : maximum iteration of the algorithm
- $\varepsilon_{\text{cond}}$ = 10^{-3} : term added to invert Hessian matrices

The last step of the algorithm consist in reprojecting the infor-

mation from the patch domain to the pixel domain. We use for that purpose the usual uniform reprojection [16].

We have conducted experiments both on simulated data (cf. Saturn image Fig. 1) and on real data (cf. Fig. 2). The last image is a single spectral channel (the 65th) of a supernova explosion in the Milky Way of the supernova remnant G1.9+0.3 (@ NASA/CXC/SAO). For this image we have access to 256 spectral channels. Performing an average across these channels provide a possible approximation of the real geometry one channel data.

In terms of PSNR for simulated data, our method globally outperforms other state-of-the-art methods such as Poisson-NLM [17], SAFIR [7], HaarTIApprox [13]. Moreover, visual artifacts tend to be reduced by our Poisson NL-PCA.

4.1. Anscombe approach and classical PCA

To compare the importance of fully taking advantage of the Poisson model and not only using the Anscombe's trick , we derive the algorithm in the Gaussian setting. It corresponds to an implementation similar to the classical power method for computing PCA [11]. The function L to be optimized in (4) is replaced by the square loss $\tilde{L}$,

$$\tilde{L}(U, V) = \sum_{i=1}^{M} \sum_{j=1}^{N} ((UV)_{i,j} - Y_{i,j})^2 \quad . \tag{10}$$

The following update equations are substituted to (8) and (9)

$$\begin{cases} U_{t+1,i,:} = U_{t,i,:} - ((U_tV_t)_{i,:} - Y_{i,:})V_t^\top (V_tV_t^\top)^{-1} , \\ V_{t+1,:,j} = V_{t,:,j} - (U_{t+1}^\top U_{t+1})^{-1}U_{t+1}^\top ((U_{t+1}V_t)_{:,j} - Y_{:,j}) . \end{cases} \tag{11}$$

An illustration of the improvement due to the direct modeling instead of a simpler Anscombe Gaussian NL-PCA approach is shown in Tab. 1. The gap is most noticeable when the noise is strong (more than 1dB increase for peak 0.1). Moreover, high-frequency artifacts are more likely to appear when using the Anscombe transform.

5. CONCLUSION AND FUTURE WORK

Inspired by the methodology of [6] we have adapted a generalization of the PCA [11, 15] for denoising images damaged by Poisson noise. Numerical, as well as visual results support the efficiency of our method, especially in the case of low count photons.

Ongoing work includes locally adapting the number of dictionary elements used to represent each denoised patches.

Peak	Direct-2P	Ansc-2P	Direct-1P	Ansc-1P	[13]
0.1	**20.31**	18.73	19.18	18.97	19.48
0.2	**22.49**	20.74	22.04	21.55	20.93
0.5	**25.58**	23.98	25.34	24.31	23.69
1	26.79	26.17	**26.88**	26.46	25.07
2	27.90	28.25	28.10	**28.34**	26.67
3	28.57	28.94	28.73	**29.12**	27.62
4	29.21	29.31	29.27	**29.61**	28.19

Table 1. PSNR comparison on the Saturn image (averaging over ten noise realizations). We compare our Poisson NL-PCA with one or two pass (Direct-1P and Direct-2P), the Gaussian NL-PCA version of our algorithm (using Anscombe transform) with one or two pass (Ansc-1P and Ansc-2P) and the haarTIApprox algorithm [13].

6. REFERENCES

[1] A. Buades, B. Coll, and J-M. Morel, "A review of image denoising algorithms, with a new one," *Multiscale Model. Simul.*, vol. 4, no. 2, pp. 490–530, 2005.

[2] K. Dabov, A. Foi, V. Katkovnik, and K. O. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, 2007.

[3] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Non-local sparse models for image restoration," in *ICCV*, 2009, pp. 2272–2279.

[4] D. D. Muresan and T. W. Parks, "Adaptive principal components and image denoising," in *ICIP*, 2003, pp. 101–104.

[5] L. Zhang, W. Dong, D. Zhang, and G. Shi, "Two-stage image denoising by principal component analysis with local pixel grouping," *Pattern Recogn.*, vol. 43, no. 4, pp. 1531–1549, 2010.

[6] C.-A. Deledalle, J. Salmon, and A. S. Dalalyan, "Image denoising with patch based PCA: Local versus global," in *BMVC*, 2011.

[7] J. Boulanger, C. Kervrann, P. Bouthemy, P. Elbau, J.-B Sibarita, and J. Salamero, "Patch-based nonlocal functional for denoising fluorescence microscopy image sequences.," *IEEE Trans. Med. Imag.*, vol. 29, no. 2, pp. 442–454, 2010.

[8] B. Zhang, J. Fadili, and J-L. Starck, "Wavelets, ridgelets, and curvelets for Poisson noise removal," *IEEE Trans. Image Process.*, vol. 17, no. 7, pp. 1093–1108, 2008.

[9] F. J. Anscombe, "The transformation of Poisson, binomial and negative-binomial data," *Biometrika*, vol. 35, pp. 246–254, 1948.

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

[11] M. Collins, S. Dasgupta, and R. E. Schapire, "A generalization of principal components analysis to the exponential family," in *NIPS*, 2002, pp. 617–624.

[12] A.P. Singh and G.J. Gordon, "A unified view of matrix factorization models," *Machine Learning and Knowledge Discovery in Databases*, pp. 358–373, 2008.

[13] R. Willett and R. Nowak, "Platelets: A multiscale approach for recovering edges and surfaces in photon-limited medical imaging," *IEEE Trans. Med. Imag.*, vol. 22, no. 3, pp. 332–350, 2003.

[14] G.J. Gordon, "Generalized^2 linear^2 models," in *NIPS*, 2003, pp. 593–600.

[15] N. Roy, G. Gordon, and S. Thrun, "Finding approximate POMDP solutions through belief compression," *Journal of Artificial Intelligence Research*, vol. 23, no. 1, pp. 1–40, 2005.

[16] J. Salmon and Y. Strozecki, "Patch reprojections for Non Local methods," *Signal Processing*, vol. In press, 2011.

[17] C.-A. Deledalle, L. Denis, and F. Tupin, "Poisson NL means: Unsupervised non local means for Poisson noise," in *ICIP*, 2010, pp. 801–804.
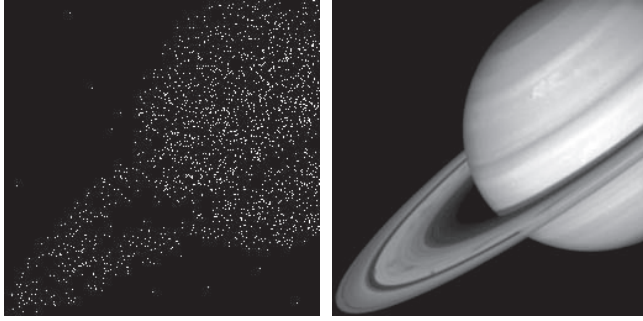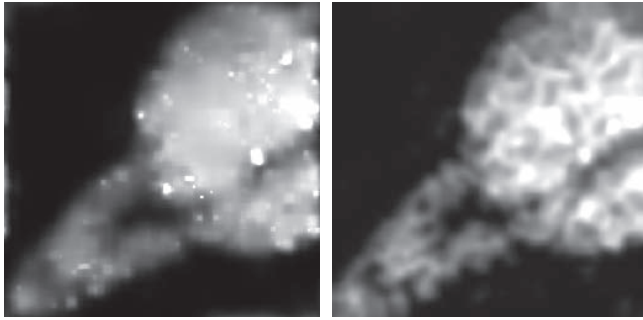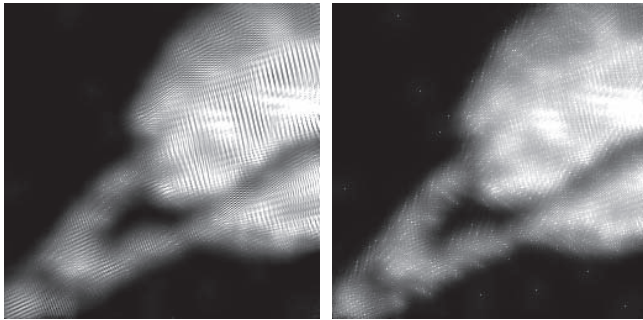
(a) Noisy image, PSNR=4.77

(b) Original image
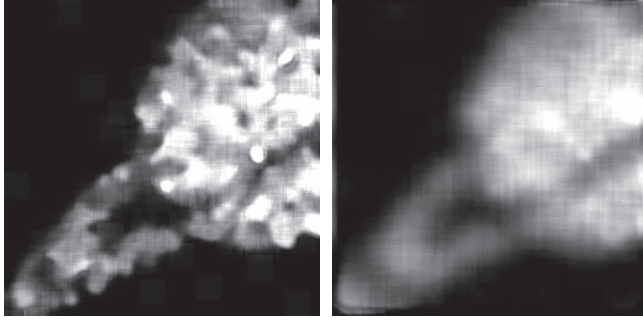
(c) PLNM, PSNR=16.34

(d) SAFIR, PSNR=19.94
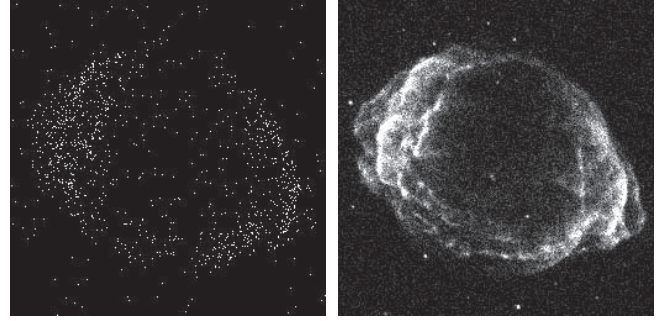
(e) HaarTIApprox, PSNR=19.23

(f) BM3D, PSNR=18.76

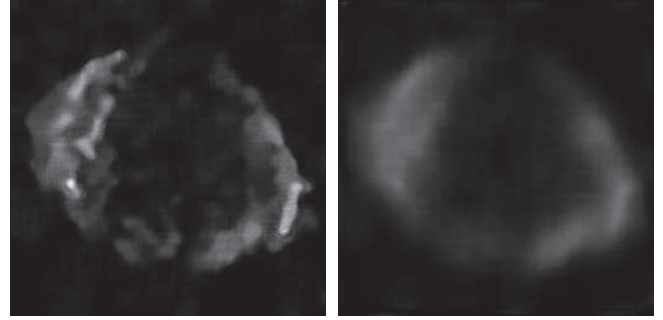(g) Gaussian NL-PCA, PSNR=17.84

(h) Poisson NL-PCA, PSNR=20.06

Fig. 1. Simulation data: comparing our Poisson NL-PCA with two iterations on the Saturn image, with Poisson NLM [17], SAFIR [7], haarTIApprox [13] and BM3D [10]. Gaussian NL-PCA, SAFIR, and BM3D use the optimized inverse Anscombe function as in [10].
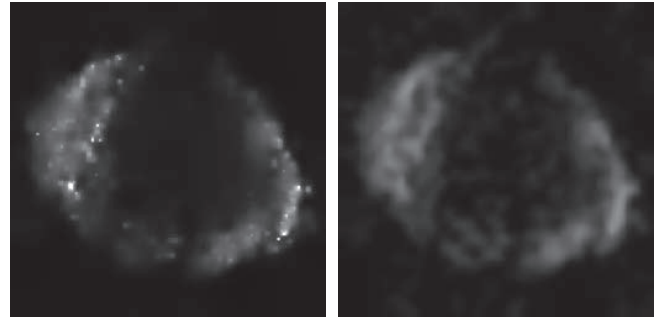

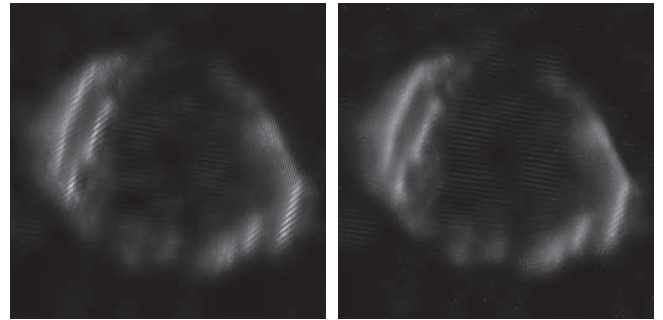(a) Single spectral channel

(b) Averaged spectral channels

(c) PLNM

(d) SAFIR

(e) HaarTIApprox

(f) BM3D

(g) Gaussian NL-PCA

(h) Poisson NL-PCA

Fig. 2. Real data: comparing our Poisson NL-PCA with two iterations on the Chandra image, with Poisson NLM [17], SAFIR [7], haarTIApprox [13] and BM3D [10]. Gaussian NL-PCA, SAFIR, and BM3D use the optimized inverse Anscombe function as in [10].