

Poisson noise reduction with Non-Local PCA (NLPCA)

Joseph Salmon, Duke University

Joint work with

Charles-Alban Deledalle (Dauphine Paris IX)

Rebecca Willett, Zachary Harmany (Duke University)

Charles



Becca



Zac



Denoising images : the Gaussian case

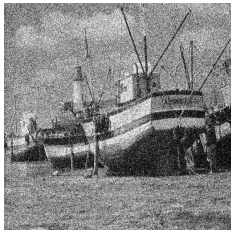
Additive White Gaussian Noise



Observed image : y

Denoising images : the Gaussian case

Additive White Gaussian Noise



Observed image : y

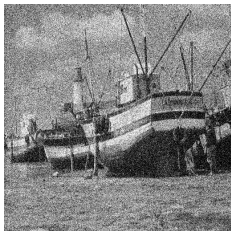
=



Ideal image : f

Denoising images : the Gaussian case

Additive White Gaussian Noise



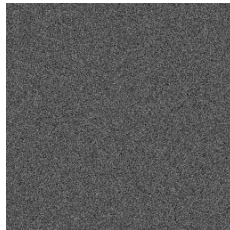
Observed image : y

=



Ideal image : f

+



Noise : ε

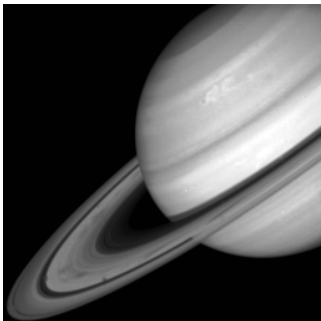
Notation :

$$y = f + \varepsilon$$

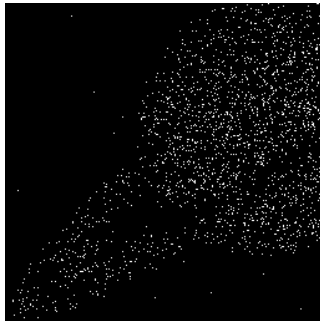
- ▶ ε : Centered Gaussian vector with known variance $\sigma^2 I$
- ▶ The image y has M pixels : $y = (y_i)_{i=1,\dots,M}$

Denoising images : the Poisson case

Poisson Noise



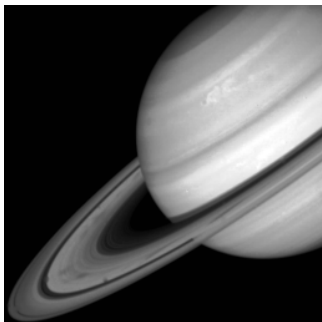
Ideal image : f



Observed image : y (Peak = 0.1)

Denoising images : the Poisson case

Poisson Noise



Ideal image : f



Observed image : y (Peak = 0.1)

Notation : $\mathbb{P}(y_i|f_i) = \frac{f_i^{y_i} e^{-f_i}}{y_i!}$ for $i = 1, \dots, M$

REM : Variance is signal dependent (increases with intensity)

Real data

FIGURE: Youngest supernova explosion in the Milky Way, supernova remnant G1.9+0.3 (@ NASA/CXC/SAO)

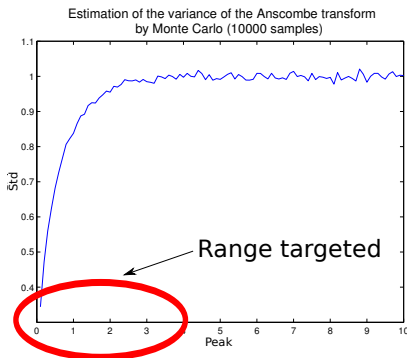
Thanks : Steven Reynolds, NC State

From Gaussian to Poisson for free ?

Variance stabilization-Anscombe transform Anscombe [48]

$$\mathbb{P}(y_i|f_i) = \frac{f_i^{y_i} e^{-f_i}}{y_i!} \quad \text{for } i = 1, \dots, M.$$

- After the transform $F : y \mapsto 2\sqrt{y + \frac{3}{8}}$, $F(y)$ is (asymptotically) close to Gaussian with unit variance



From Gaussian to Poisson for free ?

$$\mathbb{P}(y_i|f_i) = \frac{f_i^{y_i} e^{-f_i}}{y_i!} \quad \text{for } i = 1, \dots, M.$$

The whole “Anscombe” scheme

- ▶ Use Anscombe's transform $F : y_i \mapsto 2\sqrt{y_i + \frac{3}{8}}$
- ▶ Denoise as if $F(y)$ was Gaussian with unit variance
- ▶ Invert Anscombe's transform using F^{-1} or other functions
Makitalo and Foi [11]

Limitations of Anscombe based methods

- ▶ Not mathematically elegant : only based on asymptotics
- ▶ Strong noise : approximation is weak for instance in the extreme case of 0/1 observation
- ▶ (Linear) Inverse problem : destroy the linearity
if $y \approx \text{Poisson}(f)$ then $F(y) \approx \mathcal{N}(\sqrt{f}, 1)$
BUT if $y \approx \text{Poisson}(Af)$, then $F(y) \approx \mathcal{N}(\sqrt{Af}, 1)$

From Gaussian to Poisson for free ?

$$\mathbb{P}(y_i|f_i) = \frac{f_i^{y_i} e^{-f_i}}{y_i!} \quad \text{for } i = 1, \dots, M.$$

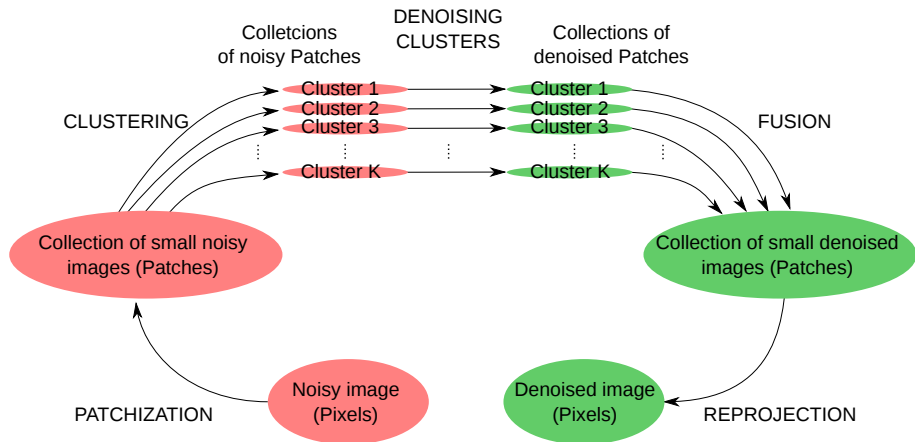
The whole “Anscombe” scheme

- ▶ Use Anscombe's transform $F : y_i \mapsto 2\sqrt{y_i + \frac{3}{8}}$
- ▶ Denoise as if $F(y)$ was Gaussian with unit variance
- ▶ Invert Anscombe's transform using F^{-1} or other functions
Makitalo and Foi [11]

Limitations of Anscombe based methods

- ▶ Not mathematically elegant : only based on asymptotics
- ▶ Strong noise : approximation is weak for instance in the extreme case of 0/1 observation
- ▶ (Linear) Inverse problem : destroy the linearity
if $y \approx \text{Poisson}(f)$ then $F(y) \approx \mathcal{N}(\sqrt{f}, 1)$
BUT if $y \approx \text{Poisson}(Af)$, then $F(y) \approx \mathcal{N}(\sqrt{Af}, 1)$

Patch based denoising



Patchization

Image

Patch

Vectorized
Patch

Collection of patches $Y \in \mathbb{R}^{N \times M}$

Clustering with K-means

K-means / E-M Algorithm

INPUT: data, number of clusters

Initialization: random clustering

while not done:

 Step 1 (E-step):

 compute for each cluster the centre (mean)

 Step 2 (M-step):

 affect each data point to the closest cluster/center

OUTPUT: K clusters of data points

REM : closeness measured through a Bregman divergence

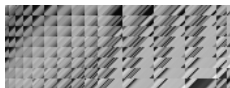
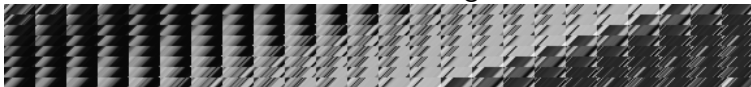
Banerjee et al. [05] (Kmeans : $d(x, x_C) = \|x - x_C\|_2^2/2$, Poisson

Kmeans : cf. later)

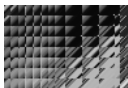
On going example

- ▶ number of clusters : $K = 5$
- ▶ number of observations : $M = 1024 = 32 \times 32$
- ▶ dimension of each patch : $N = 64 = 8 \times 8$ (patch size)

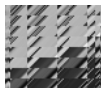
“Patchized” image



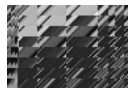
(a)



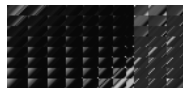
(b)



(c)



(d)



(e)

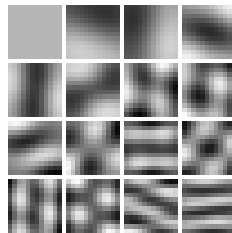
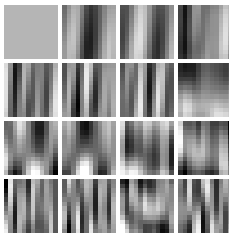
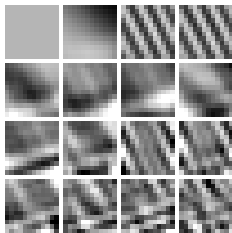
Clusters of patches

REM : for strong Poisson noise, the size of patches might be big
(e.g. 20×20)

Some denoising method using patches

- ▶ NLM Buades et al. [05]: averaging of similar patches
- ▶ BM3D Dabov et al. [07]: wavelet denoising of the clusters
- ▶ KSVD Aharon et al. [06], NLSM Mairal et al. [09]: dictionary learning.
- ▶ PCA Hirakawa and Parks [06], Zhang et al. [10], Deledalle et al. [11], Buades et al. [12]: Principal Component Analysis.

Examples of axis extracted through PCA



Low rank factorization : Gaussian case

Y : matrix representing the patches

U : dictionary

V : coefficients

Matrix formulation : approximation of order ℓ

$$(\hat{U}_\ell, \hat{V}_\ell) = \arg \min_{(U, V) \in \mathbb{R}^{N \times \ell} \times \mathbb{R}^{\ell \times M}} \|Y - UV\|_F^2$$

$$\hat{Y}_\ell = \hat{U}_\ell \hat{V}_\ell$$

REM : $D(Y \| UV) = \|Y - UV\|_F^2$ is a Bregman divergence

REM : Solution is just the truncated SVD (Singular Value Decomposition) to order ℓ

Low rank factorization : Poisson case

For the Poisson case another Bregman divergence should be used :

$$D(Y \| UV) = \sum_{i=1}^M \sum_{j=1}^N \exp(UV)_{i,j} - Y_{i,j}(UV)_{i,j}$$

Matrix formulation : approximation of order ℓ

$$(\hat{U}_\ell, \hat{V}_\ell) = \arg \min_{(U, V) \in \mathbb{R}^{N \times \ell} \times \mathbb{R}^{\ell \times M}} \sum_{i=1}^M \sum_{j=1}^N \exp(UV)_{i,j} - Y_{i,j}(UV)_{i,j}$$

$$\hat{Y}_\ell = \exp(\hat{U}_\ell \hat{V}_\ell)$$

Collins et al. [02], Singh and Gordon [08a,08b]

REM : divergence relies on the Kullback-Leibler divergence for Poisson distribution

REM : possibly use an ℓ_1 constraint on the coefficients V

Harmany et al. [12]

Practical implementation

$$(\hat{U}_\ell, \hat{V}_\ell) = \arg \min_{(U, V) \in \mathbb{R}^{N \times \ell} \times \mathbb{R}^{\ell \times M}} D(Y \| UV)$$

Alternate optimization over U , (with V being fixed) and over V (U being fixed), both problems are convex

$$\begin{cases} \hat{V}_\ell = \arg \min_{V \in \mathbb{R}^{\ell \times M}} D(Y \| UV) \\ \hat{U}_\ell = \arg \min_{U \in \mathbb{R}^{N \times \ell}} D(Y \| UV) \end{cases}$$

Newton's update (as in Gordon [03])

Initialize U_0, V_0 randomly

$$V_{t+1} = V_t - [\nabla_V^2 D(Y \| U_t V_t)]^{-1} \nabla_V D(Y \| U_t V_t)$$

$$U_{t+1} = U_t - [\nabla_U^2 D(Y \| U_t V_{t+1})]^{-1} \nabla_U D(Y \| U_t V_{t+1})$$

Practical implementation

$$(\hat{U}_\ell, \hat{V}_\ell) = \arg \min_{(U, V) \in \mathbb{R}^{N \times \ell} \times \mathbb{R}^{\ell \times M}} D(Y \| UV)$$

Alternate optimization over U , (with V being fixed) and over V (U being fixed), both problems are convex

$$\begin{cases} \hat{V}_\ell = \arg \min_{V \in \mathbb{R}^{\ell \times M}} D(Y \| UV) + \lambda \text{Pen}(V) \\ \hat{U}_\ell = \arg \min_{U \in \mathbb{R}^{N \times \ell}} D(Y \| UV) \end{cases}$$

SPIRAL's update (as in Harmany et al.[12])

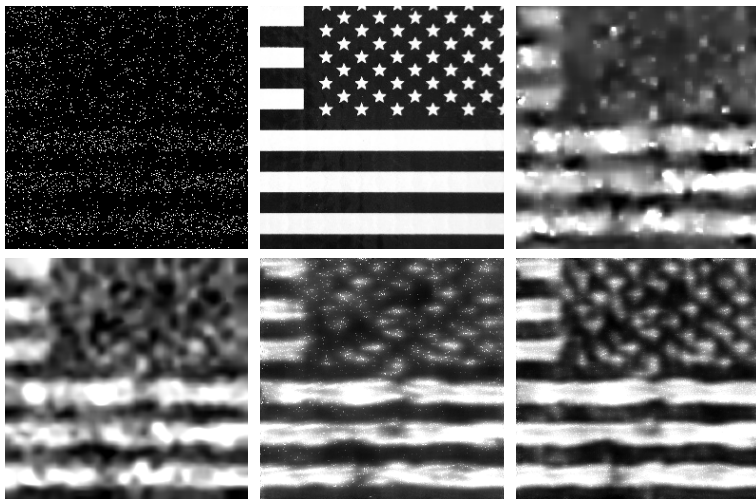
Initialize U_0, V_0 randomly

$$V_{t+1} = \text{SPIRAL}(U_t V_t, Y, \lambda, \text{Pen})$$

$$U_{t+1} = U_t - [\nabla_U^2 D(Y \| U_t V_{t+1})]^{-1} \nabla_U D(Y \| U_t V_{t+1})$$

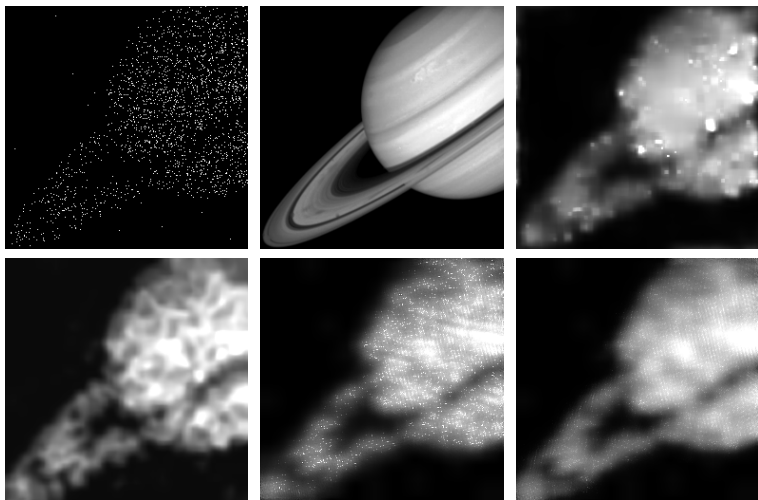
In practice we use an ℓ_1 penalty for the patch coefficients

Visual results



Noisy ,peak = 0.1 and PSNR=-7.11, original image, haarTlApprox (10.97) Willet and Nowak [03], BM3D (12.92) Makilato and Foi [11] Gaussian NLPCA (13.18) and Poisson NLPCA (14.35)

Visual results



Noisy ,peak = 0.1 and PSNR=-1.70 , original image, haarTIAprox (21.84) Willet and Nowak [03], BM3D (21.85) Makilato and Foi [11], Gaussian NLPCA (21.84) and Poisson NLPCA (22.96)

Moffet Field data set : $256 \times 256 \times 128$

In order : original, noisy (0.0387 photon per voxel), BM4D
Maggioni et al. [11], adaptive partitionning Krishnamurthy et al.
[10], NLPCAS (3x3x15), NLPCAS (5x5x23)

Conclusion

Next steps

- ▶ Speeding up the algorithm
- ▶ Extension to other noise model
- ▶ Adaptively choosing : patch size / number of atoms/ number of clusters

More information

Short version paper :

Poisson noise reduction with non-local PCA,
Salmon, Deledalle, Willett and Harmany
ICASSP 2012

Long version paper : to be posted before May 31st

Paper, slides and code available online : <http://josephsalmon.eu/>

References I