Poisson Image Deblurring

Deepan Das, Sek Cheong

University of Wisconsin, Madison

March 13 2019

Image Degradation

The Image Noise Model is one of the most important image degradation models, apart from motion blur and defocus.

Poisson Noise: This unavoidable noise is caused by the quantum nature of light.

Blur: This is caused by the natural extent of the point spread function of the imaging device.

Goal: Restore a high quality image from blurred images corrupted by Poisson Noise.

Poisson Noise Model

The various sources of light generally have random fluctuation of photons leading to spatial and temporal randomness. Mathematically, it can be represented as:

$$P(f_{(pi)} = k) = \frac{\lambda_i^k e^{-\lambda}}{k!}$$

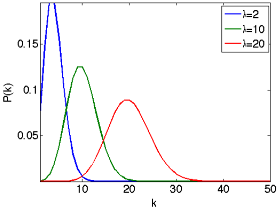


Figure: Poisson Distribution

Poisson Noise Model

Figure: Poisson Image addition

Figure: Difference

Degraded Image

A degraded image can be expressed as:

$$f = P(Hu)$$

Challenge: Linear operator H is usually ill-conditioned. Therefore, deblurring is an ill-posed inverse problem.

Previous methods: Mainly use regularization methods to restrict solution space. Some methods used previously:

- Total Variation(TV) : Preserves edges well
- Wavelets and Frames: Multi-scale and sparse representation
- Hybrid methods

Motivation for Patch Priors

- TV generally oversmooths textured images
- Wavelet based methods usually involved fixed dictionaries
- **Patch based priors** generally capture the distinctive characteristics of the processed image
- **K-SVD** based approaches learn an optimal over-complete dictionary and then recover each image patch using a linear combination of only a few atoms

Note: Poisson noise is signal dependent and Gaussian based models may not be suitable.

Assumption: Values of observed image at locations i are independent.

$$P(f|Hu) = \prod_i \frac{e_i^{-(Hu)}((Hu)_i)^{f_i}}{f_i!}$$

Some of the methods are:

- Recover image simply using methods designed for Gaussian removal
- Transform Poisson noise to near Gaussian using an appropriate transform (**Variance Stabilizing Transform**)
- Remove Poisson noise directly via a data fidelity term derived from Poisson noise statistics

Data Fidelity Approach

Bayesian Overview

Traditional Sparse representation problems were solved using:

$$min\frac{1}{2}||D\alpha - y||_2^2 + ||\alpha||_p^p$$

While attempting to remove the Poisson noise, a Bayesian approach like the previous one uses a Total Variation term and a fidelity term

$$min||\nabla u||_1 + \lambda\langle Hu - f\,logHu, 1\rangle$$

Various methods were proposed to solve this non quadratic function.

Total Variation

An Introduction

Principle: Signals with excessive and possibly spurious detail have high total variation.

$$V(x) = \sum_{i,j} \sqrt{|x_{i+1,j} - x_{i,j}|^2 + |x_{i,j+1} - x_{i,j}|^2}$$

The denoising problem still remains the same, but with a modified regularization term dependent on the TV present.

$$\min_{y} \; E(x, y) + \lambda V(x)$$

K-SVD for Gaussian Noise
Quick Recap

The fundamental assumption of the K-SVD approach is that image patches admit a sparse representation. For Gaussian noise, we have:

$$\min \sum_{(i,j)\in\mathcal{A}_n} \mu_{ij}||\alpha_{ij}||_0 + \sum_{(i,j)\in\mathcal{A}_n} ||D\alpha_{ij} - R_{ij}u||_2^2 + \lambda||u-g||_2^2$$

It was also show that learning the dictionary from noise image leads to better performance.

K-SVD for Gaussian Noise
Quick Recap

Optimization Steps

Elad et al. minimized the expression with respect to D, α_{ij} and u separately

- Solve for D given α_{ij}. Update is done one atom at a time
- Use OMP to get an efficient solution for α_{ij}
- Solve for u given D and $\alpha_{i,j}$

The K-SVD algorithm provides very good results for Gaussian noise removal and has been generalised for other applications like inpainting, demosaicking, etc.

Proposed Model
Background

In general denosing a image can be modeled by solving the minimization of loss function:

$$L(x) = \frac{1}{2}||x - y||^2 + G(x)$$

Where y is a observed image, and x is the unknown ideal image to be recovered

$$\frac{1}{2}||x - y||^2 \quad \text{measure of similarity (sanity check)}$$

$$G(x) \quad \text{is the prior or regularization}$$

The goal is to come up with a **good** prior to model the image and its noise distribution. In Bayesian view this is the equivalent of Maximum-A-Posteriori **MAP** estimation

Proposed Model

Background

- Suppose we are given a image

$$y = Hx + v$$

  Which is nosiy and degraded version of x
- How do we go about solving x?

Proposed Model
Background

- Suppose we are given a image

$$y = Hx + v$$

  Which is nosiy and degraded version of x
- How do we go about solving x?
- Assume x from $D\alpha$

Proposed Model

Background

- Suppose we are given a image

$$y = Hx + v$$

  Which is nosiy and degraded version of x
- How do we go about solving x?
- Assume x from $D\alpha$
- How about find the α that generated y?

Proposed Model

Background

With sparse model, given a noisy image $y = Hx + v$ [1] we can denoise it by solving:

$$\hat{\alpha} = \arg\min_{\alpha} ||\alpha||_0 \quad \text{s.t.} \quad ||y - D\alpha|| \leq \epsilon$$

$$\hat{x} = D\hat{\alpha}$$

Why does it work?

$$||\hat{\alpha}||_0 < ||\alpha||_0 \implies \hat{\alpha} = \alpha$$

[1] in inprinting problem H is a lost sample W the $\hat{\alpha} = \arg\min_{\alpha} ||\alpha||_0 \quad \text{s.t.} \quad ||y - WD\alpha|| \leq \epsilon$

However, we can't train a D for the entire image using K-SVD
Enforce shift-invariant sparsity, using NxN patches with overlaps:

$$\min \sum_{(i,j)\in\mathcal{A}_n} \mu_{ij}||\alpha_{ij}||_0 + \sum_{(i,j)\in\mathcal{A}_n} ||D\alpha_{ij} - R_{ij}u||_2^2 + \lambda||u-g||_2^2$$

Proposed Model

General Method

The general method of denosing an image:

1. Every pixel in $y = Hx + v$ as the center of a patch p_{ij} size $\sqrt{N} \times \sqrt{N}$ (typical value for n is 64)
2. Use apply denosing algorithm on each patch p_{ij}
3. Merge the denoised patches by averaging one on top of the other. Various averaging methods can be used.

Proposed Model

Reasoning

- Images are structure data of repeated patterns, edges, smooth regions, and texture
- Incorporating sparse and adaptive patch priors (Elad et al.)
- Use sparse prior and data-fidelity terms to model blur and Poisson noise
- Sparse representation prior as regularization term

$$\hat{\alpha} = \arg\min_{\alpha \in \mathbb{R}^K} ||\alpha||_0, \quad s.t. ||D\alpha - x||_2 \leq \epsilon$$

- Data-fidelity term: $\lambda \langle Hu - f\, logHu, 1 \rangle$
- TV term to overcome artifacts caused by patch-based priors in deblurring tasks: $||\nabla u||_1$

The Model - Putting it all together, we have :

$$\begin{aligned} \min \sum_{(i,j)\in\mathcal{A}_s} \mu_{ij}||\alpha_{ij}||_0 + \sum_{(i,j)\in\mathcal{A}_s} ||D\alpha_{ij} - R_{ij}u||_2^2 \\ +\eta||\nabla u||_1 \\ +\lambda\langle Hu - f\,logHu, 1\rangle \end{aligned} \tag{1}$$

$$\mathcal{A}_s = \{1, 2, ..., m - \sqrt{N} + 1\} \times \{1, 2, ..., n - \sqrt{N} + 1\}$$

α_{ij} - coefficient of patch at location (i, j)
μ_{ij} - hidden parameter at location (i, j)
R_{ij} - extracted patch from image location (i, i)
f - The nosiy or blurry image
u - Unknown ideal image
H - blur kernel
λ, η - balanced parameter for DF and TV terms

Proposed Model

Issues

The proposed model has the following issues:

- It is nonconvex ($D\alpha_{ij}$)
- TV regularization term is nondifferentiable
- Data-Fidelity term, because of the *log* is not easy to handle

To combat these issues

- Fix D mim. function w.r.t. $\{\alpha_i j\}$ and u
- TV use additional relaxation or constraint
- Variable Splitting method to handle DF term

Variable Splitting

$$\min_x F_1(Ax) + F_2(x)$$

Add auxiliary variable d:

$$\min_x F_1(d) + F_2(x) \quad \text{s.t.} \quad Ax = d$$

Add relaxation:

$$\min_x F_1(d) + \frac{\beta}{2}||Ax - d||^2 + F_2(x)$$

When $\beta \to +\infty$, it revert back to its original form

Proposed Model
Full Equation

Applying variable splitting to (1):

$$\min_{\{\alpha_{ij}\},p,q,w,u} \sum_{(i,j)\in\mathcal{A}_s} \mu_{ij}||\alpha_{ij}||_0 + \sum_{(i,j)\in\mathcal{A}_s} ||D\alpha_{ij} - R_{ij}p||_2^2 + \frac{\beta}{2}||p-u||_2^2 + \eta||q||_1 + \frac{\eta_1}{2}||\nabla u - q||_2^2 + \lambda\langle Hu - f\, logHu, 1\rangle + \frac{\gamma}{2}||w - Hu||_2^2 \quad (2)$$

Where $p \in \mathbb{R}^{mn}$,$q \in \mathbb{R}^{mn} \times \mathbb{R}^{mn}$, $w \in \mathbb{R}^{mn}$
and η_1, γ, β are positive large real values

Proposed Model

Solving Equation

Use alternating mim. algorithm

1. Solving for $\{\alpha_{ij}\}$ given u, D with respect to $\{\alpha_{ij}\}$ we have
$$\min_{\{\alpha_{ij}\}} \sum_{(i,j)\in\mathcal{A}_s} \mu_{ij}||\alpha_{ij}||_0 + \sum_{(i,j)\in\mathcal{A}_s} ||D\alpha_{ij} - R_{ij}p||_2^2$$
Use OMP method as in the K-SVD algorithm
2. Solving p given u, D we have
$$\min_{p} \sum_{(i,j)\in\mathcal{A}_s} \sum_{(i,j)\in\mathcal{A}_s} ||D\alpha_{ij} - R_{ij}p||_2^2 + \frac{\beta}{2}||p-u||_2^2$$
This is a least square problem and has a closed form solution
3. Solving for q given u, D, we have:
$$\min_{q} \eta||q||_1 + \frac{\eta_1}{2}||\nabla u - q||_2^2$$
l_1-regularized least square, use soft-thresholding

Use alternating mim. algorithm

3. cont'd pointwise soft-thresholding

$$q = shrink\left(\nabla u, \frac{\eta}{\eta 1}\right), \quad shrink(t, \tau) = \max(||t|| - \tau, 0)\frac{t}{||t||}$$

4. Solving w given u, D we have

$$\min_w \lambda\langle w - f\, logw, 1\rangle + \frac{\gamma}{2}||w - Hu||_2^2$$

w has explicit solution

5. Solving u given $\{\alpha_{ij}\}$, p, q, w, D we have

$$\min_u \frac{\beta}{2}||p - u||_2^2 + \frac{\eta_1}{2}||\nabla u - q||_2^2 + \frac{\gamma}{2}||w - Hu||_2^2$$

$$u = (\beta = \eta_1 \nabla^* \nabla + \lambda H^T H)^{-1}(\beta p + \eta \nabla^* q + \lambda H^T w)$$

Initialization: Set $u = f$, $D = DCT$. Choose η_1, λ, γ, η and β, the number of iterations for the outer (N_O) and inner (N_I) loops, the growth rates $r_\beta, r_\gamma, r_{\eta_1}$

Output: u

for $iter_O = 1$ to N_O **do**

  for $iter_I = 1$ to N_I **do**

    update $\alpha_{i,j}$ using OMP

    update p

    update q

    update w

    update u

  end for

  update Dictionary D

  $\beta = \beta \cdot r_\beta$

  $\gamma = \gamma \cdot r_\gamma$

  $\eta_1 = \eta_1 \cdot r_{\eta_1}$

end for

Experimental Results
Benchmarks

- Comparisons made to the TV-Based algorithm (PIDSplit+) and the frame based algorithm (PIDAL-FA)
- Quality of restoration results compared quantitatively by using the PSNR:

$$PSNR = 20log_{10}\frac{Peak}{\frac{1}{mn}||u^* - u||_2}$$

- Stopping criterion:

$$\frac{||u^{k+1} - u^k||_2}{||u^{k+1}||_2} < \epsilon$$

Results

Biomedical Images

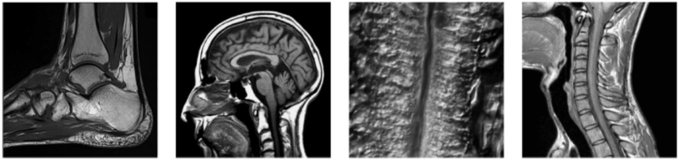
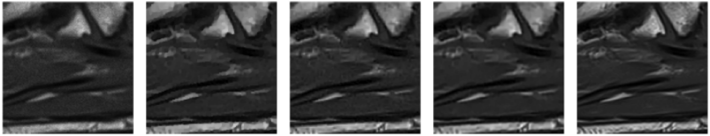


Fig. 1. Original images. From left to right: *Ankle, Brain, Mouseintestine4, Neck*.



(a) Degraded (b) PIDSplit+ (30.88dB) (c) PIDAL-FA (31.66dB) (d) Ours (32.59dB) (e) Original

Fig. 2. Results of different methods on *Ankle* image (detail) corrupted by a Gaussian blur ($\sigma = \mathbf{1}$) and Poisson noise with peak intensity 600.

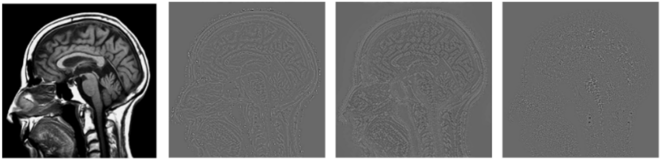
Method Noise

Evaluation metric that is adaptive to Poisson Noise. Uses the Anscombe transform.

$$A(u) - A(D_h u)$$

Applying the Anscombe transform on both leads to a Gaussian approximation.

Method Noise and Natural Images



(a) Original (b) PIDSplit+ (c) PIDAL-FA (d) Ours

Fig. 5. *Method Noise* of different methods on the *Brain* image.

Fig. 6. Original images. From left to right: *Barbara, Cameraman, Lena, Man*.

Natural Images Results

(a) Degraded (b) PIDSplit+(24.51dB) (c) PIDAL-FA (24.93dB) (d) Ours (25.29dB) (e) Original

Fig. 7. Results of different methods on *Barbara* image corrupted by the Gaussian blur and Poisson noise with peak intensity 600.

(a) Degraded (b) PIDSplit+(29.29dB) (c) PIDAL-FA (29.23dB) (d) Ours (29.82dB) (e) Original

Fig. 8. Results of different methods on *Man* image corrupted by the Gaussian blur and Poisson noise with peak intensity 600.

(a) Degraded (b) PIDSplit+(26.09dB) (c) PIDAL-FA (26.21dB) (d) Ours (26.76dB) (e) Original

Fig. 9. Results of different methods on *Cameraman* image corrupted by the uniform blur and Poisson noise with peak intensity 600.

Tabular Results

Natural Images

	Gaussian blur			Uniform blur		
Images/Peak	[28]	[14]	Ours	[28]	[14]	Ours
Bar./1000	24.98	25.46	**25.67**	24.24	24.52	**24.65**
Bar./600	24.51	24.93	**25.29**	23.95	24.23	**24.40**
Bar./255	24.07	24.34	**24.60**	23.57	23.79	**23.84**
Cam./1000	28.33	28.38	**28.77**	26.72	26.86	**26.97**
Cam./600	27.54	27.84	**28.40**	26.09	26.21	**26.76**
Cam./255	26.63	26.79	**27.52**	25.23	25.31	**26.06**
Lena/1000	32.70	32.98	**33.70**	30.92	31.25	**31.75**
Lena/600	32.27	32.38	**33.06**	30.44	30.70	**31.19**
Lena/255	31.10	31.15	**31.68**	29.65	29.75	**29.94**
Man/1000	29.92	29.78	**30.35**	28.12	28.17	**28.46**
Man/600	29.29	29.23	**29.82**	27.66	27.65	**27.99**
Man/255	28.34	28.25	**28.73**	26.90	26.90	**27.12**
Aver./1000	28.98	29.15	**29.62**	27.50	27.70	**27.96**
Aver./600	28.40	28.60	**29.14**	27.04	27.20	**27.59**
Aver./255	27.54	27.63	**28.13**	26.34	26.44	**26.74**

Figure: Natural Images

Learned Dictionary

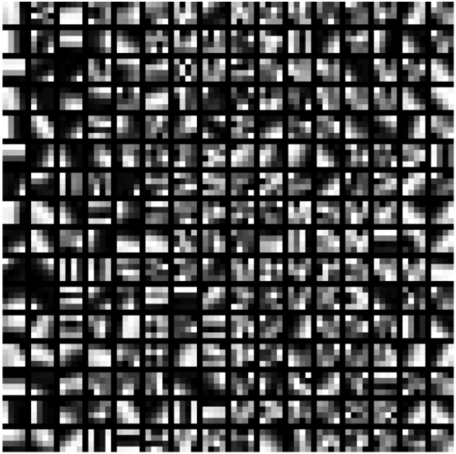


Figure: Learned Dictionary

Blind deblurring

- We assume the blur kernel H is known
- Not likely in real applications
- All existing methods used the Gaussian assumption
- Handles Poisson noise via a Variance Stabilization Transform
- Experiment includes images corrupted with a Gaussian blur and Poisson noise.

Blind Deblurring

Fig. 10. Results of different methods on *Brain* image corrupted by a Gaussian blur and Poisson noise with peak intensity 600.

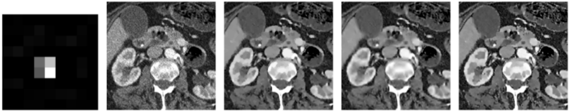


Fig. 11. Results of different methods on *CT* image.

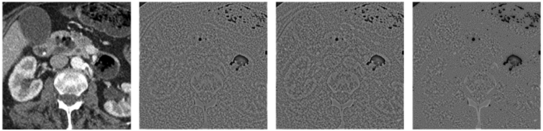


Fig. 12. *Method Noise of different methods on the CT image.*

Figure:

Multiplicative noise

- Appears in laser imaging, ultrasound, synthetic aperture radar
- Can be modeled as

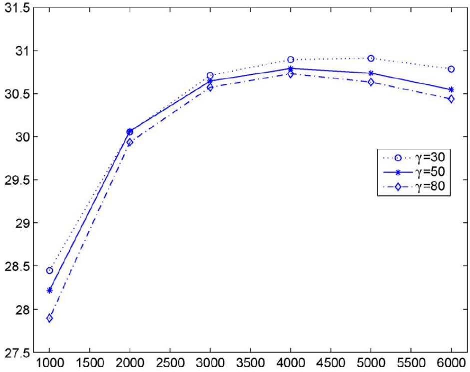
$$f = (Hu)M_n$$

  Where M_n is a random variable with mean one
- The Data-Fidelity prior still appropriate if M_n follows Gamma distribution

$$P(x;\theta,L) = \frac{1}{\theta^L \Gamma(L)} x^{L-1} e^{-\frac{x}{\theta}} \quad \text{for} \quad x \geq 0$$

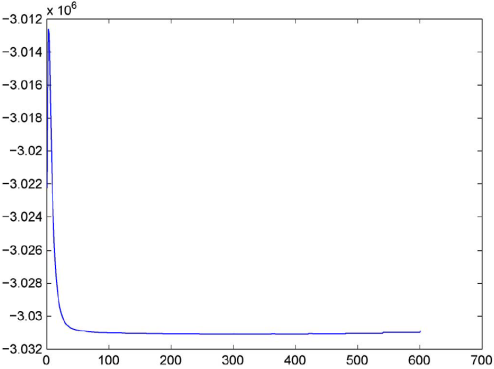
  Assume means equal to 1, so $L\theta = 1$

The Effect of the Parameters



The y axis is PSNR and x axis is the λ
η controls effect on TV, and TV also affect by ratio of η and λ

The Effect of the convergence of the Algorithm



The y axis is the error rate and x axis is the iteration
Gaussian blur and Poisson noise with peak at 600

Computation time and improvements

- Set $N_I = 60$, $N_O = 20$, 512×512 image
- Takes 10 minutes
- Having to deal with $q \in \mathbb{R}^{mn} \times \mathbb{R}^{mn}$
- PIDSplit+ and PIDAL-FA take less than 1 minute
- Use faster computer!
- Parallel computing
- Early stop from the outer loop

Conclusion

Conclusion

- Generalized Blind Deblurring
- Computation time
- Non-Gaussian extensions

Questions?