Poisson Inverse Problems by the Plug-and-Play scheme

Project submission for CS-754 (AIP)

Vivek Pawar (23m0769)
Nikita (23m0807)
M.Tech (TA), Sem-II
CSE department

Source: Rond, Arie, Raja Giryes, and Michael Elad. "Poisson inverse problems by the plug-and-play scheme." *Journal of Visual Communication and Image Representation* 41 (2016): 96-108.

Problem Statement

Traditional image restoration algorithms are designed for Gaussian noise, which is different from Poisson noise. Applying these algorithms directly to Poisson noise isn't very effective.

In an inverse problem we are given a degraded image, **y = N(H.x)** H is linear operator and N is noise operator, and want to recover from it a clean image x. The probability of getting a noisy image in poisson distribution is given by:

$$\boxed{P\left(y\left[i\right]|x\left[i\right]\right) = \begin{cases} \frac{(x[i])^{y[i]}}{y[i]!}e^{-x[i]} & \text{if } x\left[i\right] > 0 \\ \delta\left(y\left[i\right]\right) & \text{if } x\left[i\right] = 0 \end{cases}}$$

Plug and Play

The goal of the PaP framework is to maximize the posterior probability in an attempt to implement the MAP estimator.

$$\max_{x \in R^{m \times n}} P(x|y) = \max_{x \in R^{m \times n}} \frac{P(y|x)P(x)}{P(y)} = \max_{x \in R^{m \times n}} P(y|x)P(x)$$

Taking element wise −ln (·) :

$$\min_{x \in R^{m \times n}} -ln(P(x|y)) = \min_{x \in R^{m \times n}} -ln(P(y|x)) - ln(P(x))$$

Let l(x) = −ln (P (y|x)) and s (x) = −ln (P (x))

New objective will become:

$$\boxed{\hat{x} = \underset{x \in R^{m \times n}}{\arg\min}\, l(x) + \beta s(x)} \qquad s.t \;\; x = v$$

Now, this can be solved using ADMM by augmented Lagrangian given by:

$$\boxed{L_\lambda = l(x) + \beta s(v) + \frac{\lambda}{2}\|x - v + u\|_2^2 - \frac{\lambda}{2}\|u\|_2^2}$$

According to ADMM theory that minimizing $\hat{x}$ is equivalent to following these 3 steps until convergence:

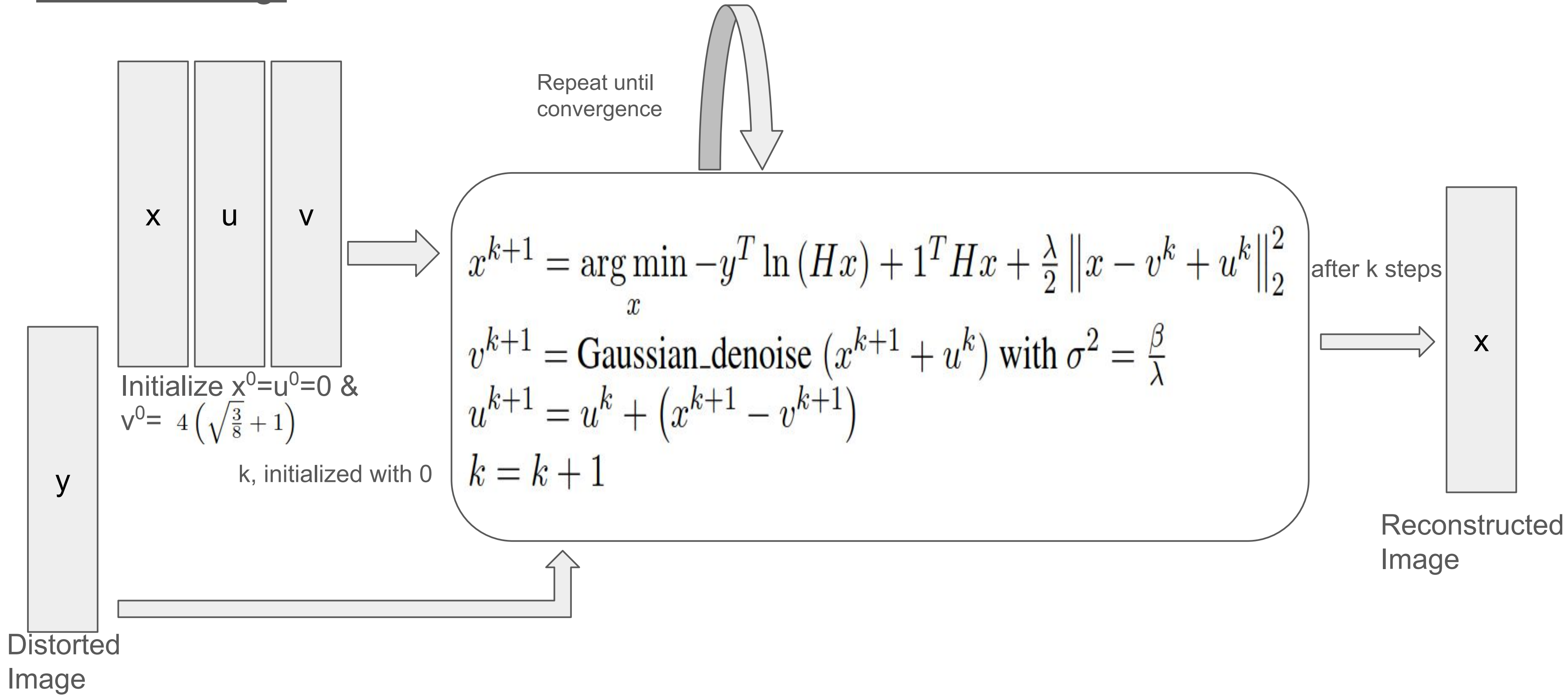
$$\begin{aligned} x^{k+1} &= \arg\min_x L_\lambda\left(x, v^k, u^k\right), \\ v^{k+1} &= \arg\min_v L_\lambda\left(x^{k+1}, v, u^k\right), \\ u^{k+1} &= u^k + \left(x^{k+1} - v^{k+1}\right). \end{aligned}$$

$$\begin{aligned} x^{k+1} &= \arg\min_x l(y|x) + \tfrac{\lambda}{2}\left\|x - \left(v^k - u^k\right)\right\|_2^2, \\ v^{k+1} &= \arg\min_v \tfrac{\lambda}{2}\left\|x^{k+1} + u^k - v\right\|_2^2 + \beta s(v), \\ u^{k+1} &= u^k + \left(x^{k+1} - v^{k+1}\right). \end{aligned}$$

Equation after substituting L_λ

Plug and Play for Poisson Inverse (P^{4}IP)

For denoising:

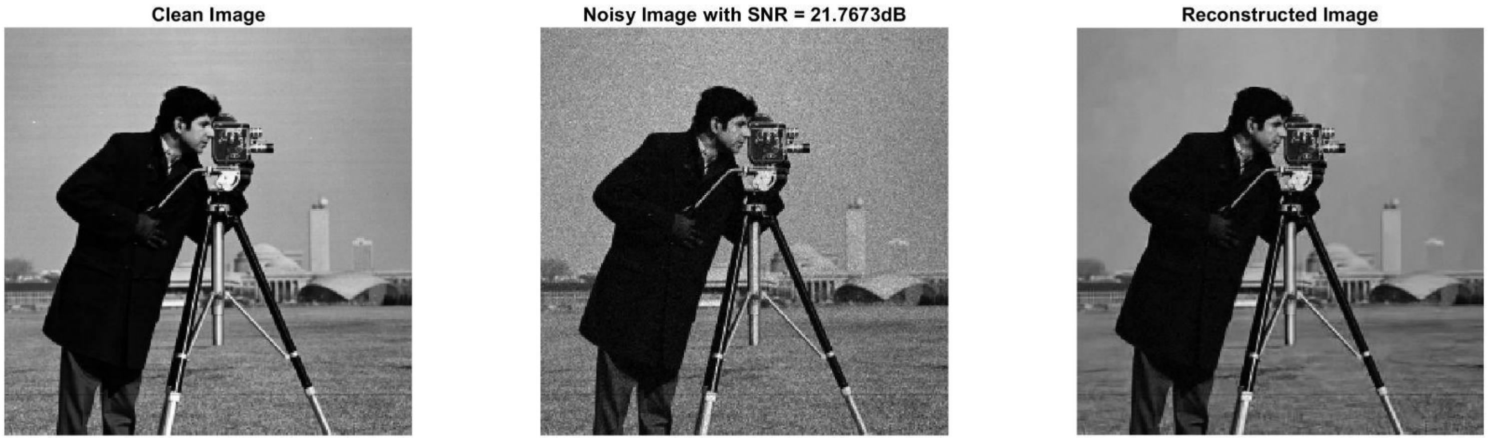


- For step 1, by taking H=I(for denoising) the closed form solution for x can be written as:

$$x^{k+1}[i] = \frac{\left(\lambda\left(v^k[i] - u^k[i]\right) - 1\right) + \sqrt{\left(\lambda\left(v^k[i] - u^k[i]\right) - 1\right)^2 + 4\lambda y[i]}}{2\lambda}$$

- BM3D is used for gaussian denoising.

<u>Results for cameraman image with λ = 0.25,</u>



For deblurring:

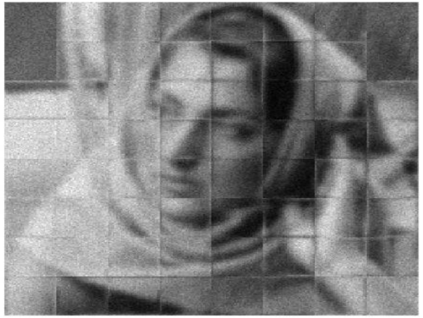
- Here H will be a blur matrix.
- For this task, blurring and also reconstruction is done patch wise.
- We update x using gradient descent with this objective function.

$$\boxed{\nabla_x L_\lambda = -H^T \left(y/\left(Hx\right)\right) + H^T 1 + \lambda \left(x - v + u\right)}$$

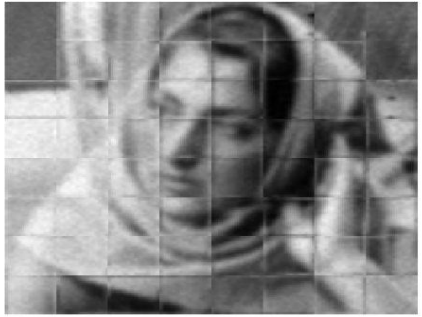
- Lbfgsb is used as a convex optimizer for above objective.
- Rest u and v are initialized and updates as done in denoising
- Also binning is done for deblurring task and the results obtained are better.
- Experiments are done for 2 kernels: Gaussian~N(0,2) and Uniform~U(mean of all pixels)

Results for deblurring with Gaussian blur kernel:

Original image



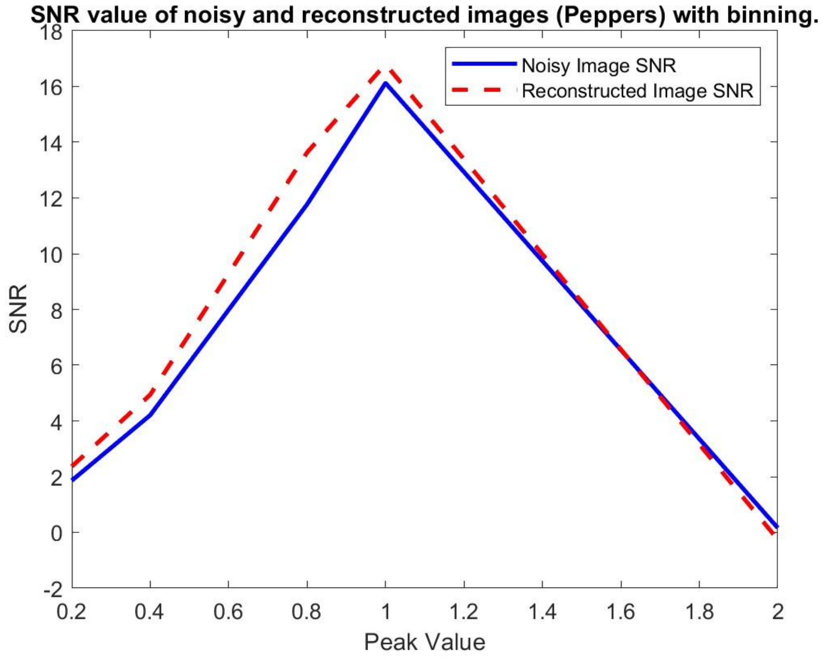
Blurred image
Peak: 1 SNR: 15.64

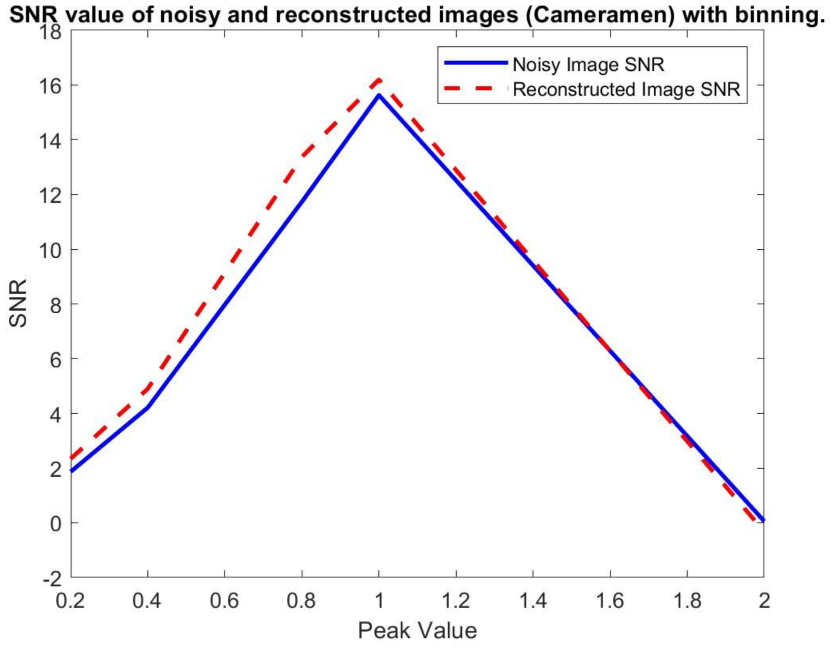


Reconstructed image
SNR: 15.72

Kernel size is 25x25

Snr value comparison:





Conclusion

- The work introduces a novel approach called "Plug-and-Play" to integrate existing Gaussian denoising algorithms into problems with Poisson noise.

- Leverages the vast availability of Gaussian denoising algorithms.

- Compared to the existing Anscombe transform approach, the new method achieves better results in cases with low noise levels ("lower peaks").

Thank You!