EE 367 / CS 448I Computational Imaging 2022
Notes: Noise, Denoising, and Image Reconstruction with Noise
Optional Notes

Gordon Wetzstein
gordon.wetzstein@stanford.edu

This document serves as a supplement to the material discussed in class. The document is not meant to be a comprehensive review of noise in imaging or denoising. It is supposed to give a concise, yet insightful summary of Gaussian and Poisson-distributed noise, denoising strategies, and deconvolution in the presence of Gaussian or Poisson-distributed noise.

In the 2022 offering of EE367, we actually won't have time to dive into this material in depth during class. Therefore, this set of notes is provided as optional for those of you interested in this topic.

1 Noise in Images

Gaussian noise combines thermal, amplifier, and read noise into a single noise term that is indenpendent of unknown signal and that stays constant for specific camera settings, such as exposure time, gain or ISO, and operating temperature. Gaussian noise parameters may change when camera settings are changed.

Signal-dependent shot noise follows a Poisson distribution and is caused by statistical quantum fluctuations in the light reaching the sensor and also in the photoelectron conversion. For example, a linear camera sensor measures irradiance – the amount of incident power of per unit area. Spectral irradiance tells us how much power was incident per unit area and per wavelength. Given some amount of measured spectral irradiance would conceptually allow us to compute the number of photons that were incident on the sensor using the definition of photon energy[1]

$$E = \frac{hc}{\lambda} \quad [J], \tag{1}$$

where E is the energy of a single photon in Joule, $h \approx 6.626 \times 10^{-34}$ [Joule $\cdot$ s] is the Planck constant, $c \approx 2.998 \times 10^8$ [m/s] is the speed of light, and λ [m] is the wavelength of light. Then, the average number of photoelectrons arriving per second per m^2 would be proportional to irradiance / (QE*photon energy), where QE is the quantum efficiency of the sensor. For scientific sensors, the manufacturer typically provides a single conversion factor CF that incorporates the pixel area and other terms, such that the number of measured photons is $\text{CF} \cdot \mathbf{b}/\text{QE}$. If this conversion factor is not available, it could be calibrated. Although it seems trivial to relate the number of photons or photoelectrons to a measured intensity value, in practice within any finite time window or exposure time, this is actually a Poisson process that needs to be modeled as such for robust image reconstruction from noisy measurements. Therefore, these notes give a brief introduction to statistical modeling of different image noise models.

2 Image Denoising with Gaussian-only Noise

A generic model for signal-independent, additive noise is

$$\mathbf{b} = \mathbf{x} + \eta, \tag{2}$$

where the noise term η follows a zero-mean i.i.d. Gaussian distribution $\eta_i \sim \mathcal{N}\left(0, \sigma^2\right)$. We can model the noise-free signal $\mathbf{x} \in \mathbb{R}^N$ as a Gaussian distribution with zero variance, i.e. $\mathbf{x}_i \sim \mathcal{N}\left(\mathbf{x}, 0\right)$, which allows us to model $\mathbf{b}$ as a

[1] https://en.wikipedia.org/wiki/Photon_energy

Gaussian distribution. Remember that the sum of two Gaussian distributions $\mathbf{y}_1 \sim \mathcal{N}\left(\mu_1, \sigma_1^2\right)$ and $\mathbf{y}_2 \sim \mathcal{N}\left(\mu_2, \sigma_2^2\right)$ is also a Gaussian distribution $\mathbf{y}_1 + \mathbf{y}_2 \sim \mathcal{N}\left(\mu_1 + \mu_2, \sigma_1^2 + \sigma_2^2\right)$. Therefore, the noisy intensity measurements $\mathbf{b}_i$ follow a per-pixel normal distribution

$$p\left(\mathbf{b}_i|\mathbf{x}_i, \sigma\right) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\left(\mathbf{b}_i - \mathbf{x}_i\right)^2}{2\sigma^2}}. \tag{3}$$

Due to the fact that the noise is independent for each pixel, we can model the joint probability of all M pixels as the product of the individual probabilities:

$$p\left(\mathbf{b}|\mathbf{x}, \sigma\right) = \prod_{i=1}^{M} p\left(\mathbf{b}_i|\mathbf{x}_i, \sigma\right) \propto e^{-\frac{\|\mathbf{b}-\mathbf{x}\|_2^2}{2\sigma^2}} \tag{4}$$

We can apply Bayes' rule to get an expression for the posterior distribution

$$p\left(\mathbf{x}|\mathbf{b}, \sigma\right) = \frac{p\left(\mathbf{b}|\mathbf{x}, \sigma\right) p\left(\mathbf{x}\right)}{p\left(\mathbf{b}\right)} \propto p\left(\mathbf{b}|\mathbf{x}, \sigma\right) p\left(\mathbf{x}\right), \tag{5}$$

where $p\left(\mathbf{x}\right)$ can be interpreted as a prior on the latent image $\mathbf{x}$. The maximum-a-posteriori estimate of $\mathbf{x}$ is usually calculated by maximizing the natural logarithm of the posterior distribution or, equivalently, minimizing the negative logarithm of the posterior distribution:

$$\begin{aligned}
\mathbf{x}_{\text{MAP}} &= \arg\max_{\{\mathbf{x}\}} \log\left(p\left(\mathbf{x}|\mathbf{b}, \sigma\right)\right) = \arg\max_{\{\mathbf{x}\}} \log\left(p\left(\mathbf{b}|\mathbf{x}, \sigma\right) p\left(\mathbf{x}\right)\right) && (6)\\
&= \arg\min_{\{\mathbf{x}\}} -\log\left(p\left(\mathbf{b}|\mathbf{x}, \sigma\right)\right) - \log\left(p\left(\mathbf{x}\right)\right) && (7)\\
&= \arg\min_{\{\mathbf{x}\}} \frac{1}{2\sigma^2} \|\mathbf{b} - \mathbf{x}\|_2^2 - \log\left(p\left(\mathbf{x}\right)\right) && (8)\\
&= \arg\min_{\{\mathbf{x}\}} \frac{1}{2\sigma^2} \|\mathbf{b} - \mathbf{x}\|_2^2 + \Psi\left(\mathbf{x}\right) && (9)
\end{aligned}$$

This derivation may seem simple, but Equation 9 is really important. This is the standard form of the objective function of a Gaussian denoising problem. This derivation provides the direct link between the statistical signal processing and the algebraic, regularized optimization perspectives of solving the denoising problem. Inspecting this derivation also tells us that with our regularized optimization approach, we always estimate the MAP solution to this problem, but we never analyze the posterior distribution. So we pick one solution from an entire distribution, but we have no information about the distribution itself, which could be very interesting!

2.1 Uniform Prior

For a uniform prior on images with normalized intensity values, i.e. $p\left(\mathbf{x}_i\right) = \begin{cases} 1 & \text{for } 0 \leq \mathbf{x}_i \leq 1 \\ 0 & \text{otherwise} \end{cases}$, this results in a maximum-likelihood estimation, which is equivalent to the common least-squared error form

$$\mathbf{x}_{flat} = \arg\min_{\{\mathbf{x}\}} \frac{1}{2\sigma^2} \|\mathbf{b} - \mathbf{x}\|_2^2 = \mathbf{b}, \tag{10}$$

resulting in a trivial closed-form solution. This means that denoising an image with additive Gaussian noise requires additional information to be imposed, via the prior $\Psi\left(\mathbf{x}\right)$.

2.2 Self-similarity Priors

Some of the most popular priors for image denoising are self-similarity priors, such as non-local means (NLM) [Buades et al. 2005] or BM3D [Dabov et al. 2007]. These were already introduced in lecture 4, but formally NLM minimizes the following objective function:

$$\mathbf{x}_{\text{NLM}} = \underset{\{\mathbf{x}\}}{\arg\min} \frac{1}{2\sigma^2} \|\mathbf{b} - \mathbf{x}\|_2^2 + \Psi(\mathbf{x}) = \text{NLM}\left(\mathbf{b}, \sigma^2\right). \tag{11}$$

Therefore, NLM (and similarly BM3D) can be interpreted not only as a (nonlinear) filter that remove noise from images, but more generally as maximum-a-posteriory solutions to the Gaussian image denoising problem outlined above. Fast implementations for both image and video denoising using NLM of grayscale and color images and videos are included in the OpenCV[2] software package. These implementations are recommended and they can be interfaced via OpenCV's Python wrappers (also from Matlab). Please see lecture 4 for qualitative and quantitative evaluations of NLM denoising.

2.3 Using Self-similarity Priors as Proximal Operators

In the notes on "Image Deconvolution with the Half-quadratic Splitting Method", we introduced the notion of proximal operators. These operators are useful for splitting-based optimization, where a challenging optimization problem is split into a set of smaller, tractable problems that can be solved in an alternating fashion. Each of these subproblems is solved via their respective proximal operator. The generic form of a proximal operator for some function $f(\mathbf{x})$ is

$$\text{prox}_{f,\rho}(\mathbf{v}) = \underset{\{\mathbf{x}\}}{\arg\min}\ f(\mathbf{x}) + \frac{\rho}{2} \|\mathbf{v} - \mathbf{x}\|_2^2 \tag{12}$$

Thus, the proximal operator for some image prior $\lambda\Psi(\mathbf{x})$ with weight λ is

$$\text{prox}_{\Psi,\rho}(\mathbf{v}) = \underset{\{\mathbf{x}\}}{\arg\min}\ \lambda\Psi(\mathbf{x}) + \frac{\rho}{2} \|\mathbf{v} - \mathbf{x}\|_2^2 \tag{13}$$

Comparing Equations 11 and 13 reveals close similarity. In fact, the MAP solution for the Gaussian denoising problem (Eq. 11) and the general proximal operator for a weighted prior (Eq. 13) are identical when $\sigma^2 = \lambda/\rho$. Thus, it turns out that we can use self-similarity based denoising methods, such as non-local means and BM3D as general image priors for all sorts of image reconstruction problems. Specifically, a generic NLM proximal operator is

$$\text{prox}_{\text{NLM},\rho}(\mathbf{v}) = \text{NLM}\left(\mathbf{v}, \sigma^2 = \frac{\lambda}{\rho}\right) \tag{14}$$

This proximal operator is most useful for image reconstruction problems, such as deconvolution and compressive imaging.

2.4 Using General Denoising Priors as Proximal Operators

In the notes on "Image Deconvolution with the Half-quadratic Splitting Method" we also saw that you can use any Gaussian denoiser as a prior, not just NLM or other self-similarity priors. Again, the proximal operator for a general denoiser is

$$\text{prox}_{\mathcal{D},\rho}(\mathbf{v}) = \text{denoise}\left(\mathbf{v}, \sigma^2 = \frac{\lambda}{\rho}\right) \tag{15}$$

[2] http://docs.opencv.org/2.4/modules/photo/doc/denoising.html

2.5 Image Reconstruction with Gaussian Noise

For linear inverse problems in imaging, Equation 2 becomes

$$\mathbf{b} = \mathbf{A}\mathbf{x} + \eta. \tag{16}$$

Following the above derivation for finding the maximum-a-posteriori solution for $\mathbf{x}$ results in the following optimization problem:

$$\mathbf{x}_{MAP} = \underset{\{\mathbf{x}\}}{\arg\min} \frac{1}{2\sigma^2} \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2^2 + \Psi(\mathbf{x}). \tag{17}$$

This is a least squares problem, which we have already encountered in lecture 6. Using ADMM, we can solve this problem efficiently by splitting the problem into two parts: the quadratic problem representing the data fit and the prior. For the specific case of deconvolution with a TV prior, ADMM results in several subproblems to be solved in an iterative fashion, where each of the subproblems has a closed-form solution that is given by the respective proximal operators (see lecture 6 notes). For the general case of $\mathbf{A}$ being a matrix that does not necessarily have an algebraic inverse, please see lecture 11 notes for more details.

3 Image Reconstruction with Poisson-only Noise

In many imaging problems, such as low-light photography, microscopy, and scientific imaging, however, Poisson-distributed Shot noise dominates the image formation. Note that no engineering effort can mitigate this, because shot noise is an inherent property of the particle nature of light. The image formation can be expressed as

$$\mathbf{b} \sim \mathcal{P}(\mathbf{A}\mathbf{x}), \tag{18}$$

where $\mathcal{P}$ models a Poisson process. The probability of having taken a measurement at one particular pixel i is given as

$$p(\mathbf{b}_i|\mathbf{x}) = \frac{(\mathbf{A}\mathbf{x})_i^{\mathbf{b}_i} e^{-(\mathbf{A}\mathbf{x})_i}}{\mathbf{b}_i!}. \tag{19}$$

Using the notational trick $u^v = e^{\log(u)v}$, the joint probability of all measurements is expressed as

$$p(\mathbf{b}|\mathbf{x}) = \prod_{i=1}^{M} p(\mathbf{b}_i|\mathbf{x}) = \prod_{i=1}^{M} e^{\log\left((\mathbf{A}\mathbf{x})_i\right)\mathbf{b}_i} \times e^{-(\mathbf{A}\mathbf{x})_i} \times \frac{1}{\mathbf{b}_i!}. \tag{20}$$

Note that it is essential to model the joint probability for this image formation, because elements in $\mathbf{x}$ may affect any or all measurements due to the mixing matrix $\mathbf{A}$. The log-likelihood of this expression is

$$\begin{aligned} \log(L(\mathbf{x})) &= \log(p(\mathbf{b}|\mathbf{x})) \\ &= \sum_{i=1}^{M} \log(\mathbf{A}\mathbf{x})_i \mathbf{b}_i - \sum_{i=1}^{M} (\mathbf{A}\mathbf{x})_i - \sum_{i=1}^{M} \log(\mathbf{b}_i!) \\ &= \log(\mathbf{A}\mathbf{x})^T \mathbf{b} - (\mathbf{A}\mathbf{x})^T \mathbf{1} - \sum_{i=1}^{M} \log(\mathbf{b}_i!) \end{aligned} \tag{21}$$

and its gradient is

$$\nabla \log(L(\mathbf{x})) = \mathbf{A}^T \text{diag}(\mathbf{A}\mathbf{x})^{-1} \mathbf{b} - \mathbf{A}^T\mathbf{1} = \mathbf{A}^T \left(\frac{\mathbf{b}}{\mathbf{A}\mathbf{x}}\right) - \mathbf{A}^T\mathbf{1}. \tag{22}$$

3.1 Richardson-Lucy (RL) Deconvolution

The most popular method to estimate an image from noisy measurements in this context is the Richardson-Lucy method [Richardson 1972; Lucy 1974]. In this algorithm, we use a simple idea: assume we have some initial guess of our unknown image, we can apply an iterative correction to the image which will be converged at iteration q if the estimate does not change after further iterations, i.e. $\mathbf{x}^{(q+1)}/\mathbf{x}^{(q)} = 1$. At this estimate, the gradient of the objective will also be zero. Thus, we can rearrange Equation 22 as

$$\text{diag}\left(\mathbf{A}^T\mathbf{1}\right)^{-1}\mathbf{A}^T\text{diag}\left(\mathbf{A}\mathbf{x}\right)^{-1}\mathbf{b} = \frac{\mathbf{A}^T\left(\frac{\mathbf{b}}{\mathbf{A}\mathbf{x}}\right)}{\mathbf{A}^T\mathbf{1}} = \mathbf{1} \tag{23}$$

and then perform the iterative, multiplicative update

$$\mathbf{x}^{(q+1)} = \left(\text{diag}\left(\mathbf{A}^T\mathbf{1}\right)^{-1}\mathbf{A}^T\text{diag}\left(\mathbf{A}\mathbf{x}\right)^{-1}\mathbf{b}\right)\cdot\mathbf{x}^{(q)} = \frac{\mathbf{A}^T\left(\frac{\mathbf{b}}{\mathbf{A}\mathbf{x}}\right)}{\mathbf{A}^T\mathbf{1}}\cdot\mathbf{x}^{(q)}, \tag{24}$$

where $\cdot$ is an element-wise multiplication and the fractions in Equations 22–24 are also defined as element-wise operations. This is the basic, iterative Richardson-Lucy method. It is so popular, because it is simple, it can easily be implemented for large-scale, matrix-free problems, and it uses a purely multiplicative update rule. For any initial guess that is purely positive, i.e. $0 \leq \mathbf{x}^{(0)}$, the following iterations will also remain positive. Non-negativity of the estimated image is an important requirement for most imaging applications.

3.1.1 Gradient Ascent Interpretation

The above expression can be derived as gradient ascent on the log-likelihood with a specific choice of step size. Taking the usual gradient ascent update

$$\mathbf{x}^{(q+1)} = \mathbf{x}^{(q)} + \alpha\cdot\nabla\log\left(L\left(\mathbf{x}\right)\right) \tag{25}$$

with the adaptive, per-element step size $\alpha = \mathbf{x}^{(q)}/\mathbf{A}^T\mathbf{1}$, we derive the exact same update:

$$\begin{aligned}
\mathbf{x}^{(q+1)} &= \mathbf{x}^{(q)} + \frac{\mathbf{x}^{(q)}}{\mathbf{A}^T\mathbf{1}}\cdot\left(\mathbf{A}^T\left(\frac{\mathbf{b}}{\mathbf{A}\mathbf{x}}\right) - \mathbf{A}^T\mathbf{1}\right)\\
&= \mathbf{x}^{(q)}\cdot\left(\mathbf{1} + \frac{\mathbf{A}^T\left(\frac{\mathbf{b}}{\mathbf{A}\mathbf{x}}\right)}{\mathbf{A}^T\mathbf{1}} - \mathbf{1}\right)\\
&= \mathbf{x}^{(q)}\cdot\left(\frac{\mathbf{A}^T\left(\frac{\mathbf{b}}{\mathbf{A}\mathbf{x}}\right)}{\mathbf{A}^T\mathbf{1}}\right).
\end{aligned}$$

While this seems like a simple manipulation, it will become useful when we add a total variation prior.

3.2 Richardson-Lucy (RL) Deconvolution with a TV Prior

One intuitive way of incorporating a simple anisotropic TV prior into the Richardson-Lucy algorithm is derived in the following. We aim for a maximum-a-posteriori estimation of the unknown image $\mathbf{x}$ while assuming that this image had sparse gradients. By adding a TV prior on the image as $p\left(\mathbf{x}\right) \propto \exp\left(-\lambda\left\|\mathbf{D}\mathbf{x}\right\|_1\right)$, the log-likelihood outlined in Equation 21 changes to

$$\log\left(L_{TV}\left(\mathbf{x}\right)\right) = \log\left(p\left(\mathbf{b}|\mathbf{x}\right)\right) + \log\left(p\left(\mathbf{x}\right)\right) = \log\left(\mathbf{A}\mathbf{x}\right)^T\mathbf{b} - \left(\mathbf{A}\mathbf{x}\right)^T\mathbf{1} - \sum_{i=1}^{M}\log\left(\mathbf{b}_i!\right) - \lambda\left\|\mathbf{D}\mathbf{x}\right\|_1, \tag{26}$$

therefore its gradient becomes

$$\nabla \log\left(L_{TV}\left(\mathbf{x}\right)\right) = \mathbf{A}^T \mathrm{diag}\left(\mathbf{A}\mathbf{x}\right)^{-1}\mathbf{b} - \mathbf{A}^T\mathbf{1} + \nabla\lambda\left\|\nabla\mathbf{x}\right\|_1 = \mathbf{A}^T\left(\frac{\mathbf{b}}{\mathbf{A}\mathbf{x}}\right) - \mathbf{A}^T\mathbf{1} - \nabla\lambda\left\|\mathbf{D}\mathbf{x}\right\|_1. \tag{27}$$

For an anisotropic TV norm the gradient can be derived as

$$\nabla\lambda\left\|\mathbf{D}\mathbf{x}\right\|_1 = \lambda\nabla\sum_{ij}\left(\sqrt{\left(\nabla_x\mathbf{x}\right)_{ij}^2} + \sqrt{\left(\nabla_y\mathbf{x}\right)_{ij}^2}\right) = \lambda\left(\mathbf{D}_x^T\frac{\mathbf{D}_x\mathbf{x}}{\left|\mathbf{D}_x\mathbf{x}\right|} + \mathbf{D}_y^T\frac{\mathbf{D}_y\mathbf{x}}{\left|\mathbf{D}_y\mathbf{x}\right|}\right). \tag{28}$$

Following the derivation of the last section, we can derive multiplicative update rules that include the TV prior as

$$\mathbf{x}^{(q+1)} = \frac{\mathbf{A}^T\left(\frac{\mathbf{b}}{\mathbf{A}\mathbf{x}}\right)}{\mathbf{A}^T\mathbf{1} + \lambda\left(\mathbf{D}_x^T\frac{\mathbf{D}_x\mathbf{x}}{\left|\mathbf{D}_x\mathbf{x}\right|} + \mathbf{D}_y^T\frac{\mathbf{D}_y\mathbf{x}}{\left|\mathbf{D}_y\mathbf{x}\right|}\right)} \cdot \mathbf{x}^{(q)}, \tag{29}$$

where $\cdot$ is an element-wise multiplication and the fractions in Equations 27–29 are also defined as element-wise operations. In this case, the step size is $\alpha = \mathbf{x}^{(q)} / \left(\mathbf{A}^T\mathbf{1} + \lambda\left(\mathbf{D}_x^T\frac{\mathbf{D}_x\mathbf{x}}{\left|\mathbf{D}_x\mathbf{x}\right|} + \mathbf{D}_y^T\frac{\mathbf{D}_y\mathbf{x}}{\left|\mathbf{D}_y\mathbf{x}\right|}\right)\right)$.

Although this looks like a simple extension of conventional RL, it is actually a little more tricky. First, the denominator is not guaranteed to remain positive anymore. So the the updates can get negative and need to be clamped back into the feasible range after every iteration. Second, the gradients in x and y $\left|\mathbf{D}_{x/y}\mathbf{x}\right|$ can be zero, in which case we divide by zero, which is usually not a good thing. In the latter case, we simply set the fraction $\frac{\mathbf{D}_{x/y}\mathbf{x}}{\left|\mathbf{D}_{x/y}\mathbf{x}\right|} = 0$ for the pixels where the gradients are zero. To address the first issue, we basically have to restrict ourselves to very small values of λ (i.e. 0.001-0.01) and hope that the denominator remains positive. More information about a version of this approach that uses the isotropic TV norm, but a similar approach can be found in [Dey et al. 2004b; Dey et al. 2004a].

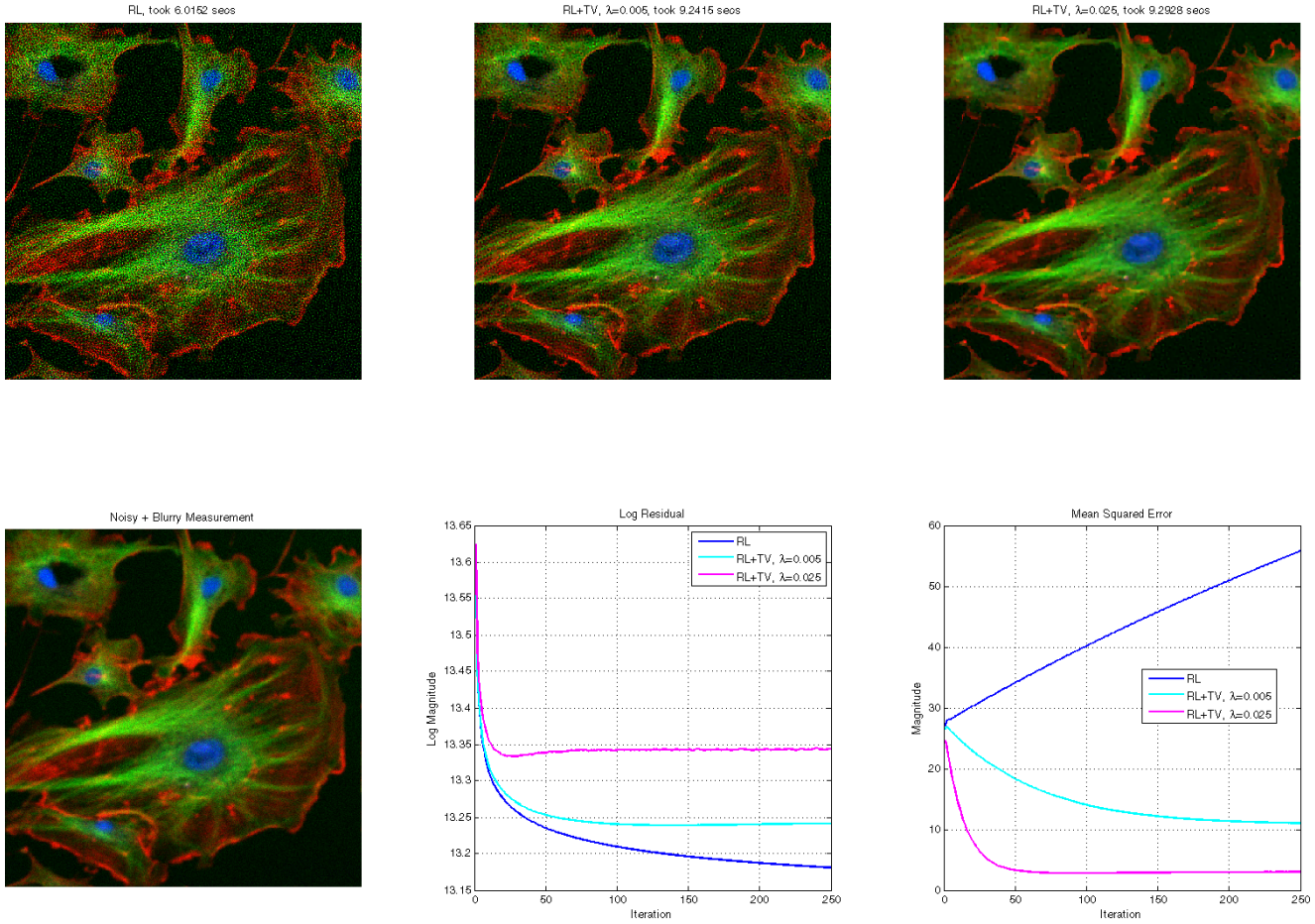

Figure 1: *We show a blurry and noisy measurement (bottom left) and the standard RL reconstruction (top left). As seen in the bottom center plot, the residual of this solution is very small and continues to decrease, but the solution is actually much too noisy. This is also seen in the mean-squared error (MSE) to the original image (bottom right). In fact, running RL for many iterations will increase the MSE and make the reconstruction worse for more iterations. Adding a TV prior will stabilize this noise-amplification behavior and result in significantly improved reconstructions (top, center and right). For a λ parameter that is too small, the solution will be close to the conventional RL solution and if the parameter is too large, the reconstruction will fail too. However, if the parameter is choosen correctly (top right), the solution looks very good.*

3.3 Poisson Deconvolution with ADMM

In this section, we derive an efficient ADMM-based solver for the deconvolution problem with Poisson noise instead of Gaussian noise. This problem could of course also be solved with the HQS method. We omit that derivation here – you can just use the ADMM solver and omit $\mathbf{u}$ to do that if you like.

Students working on research problems involving image deconvolution with Poisson noise, as observed in many scientific and bio-medical applications, may find the following derivations useful. For extended discussions, please refer to [Figueiredo and Bioucas-Dias 2010; Wen et al. 2016].

The Richardson-Lucy algorithm is intuitive and easy to implement, but it sometimes converges slowly and it is not easy to incorporate additional priors on the recovered signal. Therefore, we derive an ADMM-based formulation of the Poisson deconvolution problem to computer better solutions and to make the solver more flexible w.r.t. image priors. The ADMM formulation uses an intuitive idea: we split the Poisson deconvolution problem into several

subproblems that each can be solved efficiently, including a maximum likelihood estimator for the Poisson noise term and a separate deconvolution problem. However, we also need to take care of the non-negativity constraints on the unknown image x, which makes this problem not straightforward.

Again, the goal is to first transform our problem into the standard form of ADMM [Boyd et al. 2011], just as we did for the deconvolution problem in lecture 6. Then, we can use the iterative ADMM update rules to solve the problem using the proximal operators of the individual subproblems. Remember, the standard form of ADMM is

$$\begin{aligned} \underset{\{\mathbf{x}\}}{\text{minimize}} \quad & f(\mathbf{x}) + g(\mathbf{z}) \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} = \mathbf{c} \end{aligned} \tag{30}$$

Since we already used the symbol $\mathbf{A}$ to describe the image formation in our specific problem, we will denote that as $\mathbf{K}$ in this document. Further, the standard ADMM terms in our problem are $\mathbf{A} = \mathbf{K}$, $\mathbf{B} = -\mathbf{I}$ and $\mathbf{c} = 0$, so we will omit them in the remainder of this document (further, we already use $\mathbf{c}$ as the symbol for our PSF).

For generality, we will represent our problem as a sum of penalty functions $g_i(\cdot)$

$$\begin{aligned} \underset{\{\mathbf{x}\}}{\text{minimize}} \quad & \overbrace{\sum_{i=1}^{N} g_i(\mathbf{z}_i)}^{g(\mathbf{z})} \\ \text{subject to} \quad & \mathbf{K}\mathbf{x} - \mathbf{z} = 0, \qquad \mathbf{K} = \begin{bmatrix} \mathbf{K}_1 \\ \vdots \\ \mathbf{K}_N \end{bmatrix}, \quad \mathbf{z} = \begin{bmatrix} \mathbf{z}_1 \\ \vdots \\ \mathbf{z}_N \end{bmatrix} \end{aligned} \tag{31}$$

Although, this equation looks slightly different from the standard form of ADMM, it is actually exactly that. We just do not have an $f(\mathbf{x})$ term and we omit all the symbols we are not using.

For the Poisson deconvolution problem, we chose the following two penalty functions:

$$g_1(\mathbf{z}_1) = -\log\left(p(\mathbf{b}|\mathbf{z}_1)\right), \qquad g_2(\mathbf{z}_2) = \mathcal{I}_{\mathbb{R}_+}(\mathbf{z}_2) \tag{32}$$

Here, $g_1(\cdot)$ is the maximum likelihood estimator for the Poisson noise term and $g_2(\cdot)$ is the indicator function

$$\mathcal{I}_{\mathbb{R}_+}(z) = \begin{cases} 0 & z \in \mathbb{R}_+ \\ +\infty & z \notin \mathbb{R}_+ \end{cases} \tag{33}$$

where $\mathbb{R}_+$ is the closed nonempty convex set representing the nonnegativity constraints. The Poisson deconvolution problem written in our slight variation of the ADMM standard form then becomes

$$\underset{\{\mathbf{x}\}}{\text{minimize}} \qquad -\log\left(p(\mathbf{b}|\mathbf{z}_1)\right) + \mathcal{I}_{\mathbb{R}_+}(\mathbf{z}_2) \tag{34}$$

$$\text{subject to} \qquad \underbrace{\begin{bmatrix} \mathbf{A} \\ \mathbf{I} \end{bmatrix}}_{\mathbf{K}} \mathbf{x} - \underbrace{\begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{bmatrix}}_{\mathbf{z}} = 0 \tag{35}$$

The Augmented Lagrangian of this objective is formulated as

$$L_\rho(\mathbf{x}, \mathbf{z}, \mathbf{y}) = g_1(\mathbf{z}_1) + g_2(\mathbf{z}_2) + \mathbf{y}^T(\mathbf{K}\mathbf{x} - \mathbf{z}) + \frac{\rho}{2}\|\mathbf{K}\mathbf{x} - \mathbf{z}\|_2^2, \tag{36}$$

which leads to the following iterative updates

for $k = 1$ **to** $maxiter$

$$\mathbf{x} \leftarrow \mathbf{prox}_{quad}\left(\mathbf{v}\right) = \underset{\{\mathbf{x}\}}{\arg\min}\; L_{\rho}\left(\mathbf{x},\mathbf{z},\mathbf{y}\right) = \underset{\{\mathbf{x}\}}{\arg\min}\; \frac{1}{2}\left\|\mathbf{K}\mathbf{x}-\mathbf{v}\right\|_2^2, \quad \mathbf{v} = \mathbf{z}-\mathbf{u} \tag{37}$$

$$\mathbf{z}_1 \leftarrow \mathbf{prox}_{\mathcal{P},\rho}\left(\mathbf{v}\right) = \underset{\{\mathbf{z}_1\}}{\arg\min}\; L_{\rho}\left(\mathbf{x},\mathbf{z},\mathbf{y}\right) = \underset{\{\mathbf{z}_1\}}{\arg\min}\; -\log\left(p\left(\mathbf{b}|\mathbf{z}_1\right)\right) + \frac{\rho}{2}\left\|\mathbf{v}-\mathbf{z}_1\right\|_2^2, \quad \mathbf{v} = \mathbf{A}\mathbf{x}+\mathbf{u}_1 \tag{38}$$

$$\mathbf{z}_2 \leftarrow \mathbf{prox}_{\mathcal{I},\rho}\left(\mathbf{v}\right) = \underset{\{\mathbf{z}_2\}}{\arg\min}\; L_{\rho}\left(\mathbf{x},\mathbf{z},\mathbf{y}\right) = \underset{\{\mathbf{z}_2\}}{\arg\min}\; \mathcal{I}_{\mathbb{R}_+}\left(\mathbf{z}_2\right) + \frac{\rho}{2}\left\|\mathbf{v}-\mathbf{z}\right\|_2^2, \quad \mathbf{v} = \mathbf{x}+\mathbf{u}_2 \tag{39}$$

$$\underbrace{\begin{bmatrix}\mathbf{u}_1\\ \mathbf{u}_2\end{bmatrix}}_{\mathbf{u}} \leftarrow \mathbf{u}+\mathbf{K}\mathbf{x}-\mathbf{z}$$

end for

We derive these proximal operators for each of the updates $(\mathbf{x}, \mathbf{z}_1, \mathbf{z}_2)$ in the following.

3.3.1 Proximal Operator for Quadratic Subproblem $\mathbf{prox}_{quad}$

For the proximal operator of the quadratic subproblem (Eq. 37), we formulate the closed-form solution via the normal equations. Further, for the particular case of a deconvolution, the system matrix $\mathbf{A}$ is a convolution with the PSF c, so the proximal operator becomes

$$\begin{aligned}\mathbf{prox}_{quad}\left(\mathbf{v}\right) &= \underset{\{\mathbf{x}\}}{\arg\min}\; \frac{1}{2}\left\|\begin{bmatrix}\mathbf{A}\\ \mathbf{I}\end{bmatrix}\mathbf{x}-\begin{bmatrix}\mathbf{v}_1\\ \mathbf{v}_2\end{bmatrix}\right\|_2^2\\ &= \left(\mathbf{A}^T\mathbf{A}+\mathbf{I}\right)^{-1}\left(\mathbf{A}^T\mathbf{v}_1+\mathbf{v}_2\right)\\ &= \mathcal{F}^{-1}\left(\frac{\mathcal{F}\left\{c\right\}^*\cdot\mathcal{F}\left\{v_1\right\}+\mathcal{F}\left\{v_2\right\}}{\mathcal{F}\left\{c\right\}^*\cdot\mathcal{F}\left\{c\right\}+1}\right)\end{aligned} \tag{40}$$

Note that this proximal operator does not require explicit matrix-vector products to be computed. The structure of this specific problem allows for a closed-form solution to be expressed using a few Fourier transforms as well as element-wise multiplications and divisions. Typically, $\mathcal{F}\left\{c\right\}$ and the denominator can be precomputed and do not have to be updated throughout the ADMM iterations.

3.3.2 Proximal Operator for Maximum Likelihood Estimator of Poisson Noise $\mathbf{prox}_{\mathcal{P}}$

Recall, this proximal operator is defined (see Eq. 38) as

$$\mathbf{prox}_{\mathcal{P},\rho}\left(\mathbf{v}\right) = \underset{\{\mathbf{z}_1\}}{\arg\min}\; J\left(\mathbf{z}_1\right) = \underset{\{\mathbf{z}_1\}}{\arg\min}\; -\log\left(p\left(\mathbf{b}|\mathbf{z}_1\right)\right) + \frac{\rho}{2}\left\|\mathbf{v}-\mathbf{z}_1\right\|_2^2 \tag{41}$$

The objective function $J\left(\mathbf{z}\right)$ for the estimator is

$$J\left(\mathbf{z}_1\right) = -\log\left(\mathbf{z}_1\right)^T\mathbf{b} + \mathbf{z}_1^T\mathbf{1} + \frac{\rho}{2}\left(\mathbf{z}_1-\mathbf{v}\right)^T\left(\mathbf{z}_1-\mathbf{v}\right), \tag{42}$$

which is closely related to conventional RL. We equate the gradient of the objective to zero

$$\nabla J\left(\mathbf{z}_1\right) = -\mathrm{diag}\left(\mathbf{z}_1\right)^{-1}\mathbf{b} + \mathbf{1} + \rho\left(\mathbf{z}_1-\mathbf{v}\right) = 0. \tag{43}$$

The primary difference between this proximal operator and conventional RL is that the proximal operator removes the need for the system matrix $\mathbf{A}$. This makes the noisy observations independent of each other, so we do not need to account for the *joint probability* of all measurements. This can be seen by looking at the gradient of J w.r.t. individual elements in $\mathbf{z}_1$:

$$\frac{\partial J}{\partial \mathbf{z}_{1_j}} = -\frac{\mathbf{b}_j}{\mathbf{z}_{1_j}} + 1 + \rho\left(\mathbf{z}_{1_j} - \mathbf{v}_j\right) \tag{44}$$

$$= \mathbf{z}_{1_j}^2 + \frac{1-\rho\mathbf{v}_j}{\rho}\mathbf{z}_{1_j} - \frac{\mathbf{b}_j}{\rho} = 0 \tag{45}$$

This expression is a classical root-finding problem of a quadratic, which can be solved independently for each $\mathbf{z}_{1_j}$. The quadratic has two roots, but we are only interested in the positive one. Thus, we can define the proximal operator as

$$\mathbf{prox}_{\mathcal{P},\rho}\left(\mathbf{v}\right) = -\frac{1-\rho\mathbf{v}}{2\rho} + \sqrt{\left(\frac{1-\rho\mathbf{v}}{2\rho}\right)^2 + \frac{\mathbf{b}}{\rho}}. \tag{46}$$

This operator is independently applied to each pixel and can therefore be implemented analytically without any iterations. Note that we still need to chose a parameter ρ for the ADMM updates. Heuristically, small values (e.g. $1e^{-5}$) work best for large signals with little noise, but as the Poisson noise starts to dominate the image formation, ρ should be higher.

For more details on this proximal operator, please refer to the paper by Dupe et al. [2011].

3.3.3 Proximal Operator for Indicator Function $\mathbf{prox}_{\mathcal{I}}$

The proximal operator for the indicator function (Eq. 39) representing the constraints is rather straightforward

$$\mathbf{prox}_{\mathcal{I},\rho}\left(\mathbf{v}\right) = \underset{\{\mathbf{z}_2\}}{\arg\min}\, \mathcal{I}_{\mathbb{R}_+}\left(\mathbf{z}_2\right) + \frac{\rho}{2}\left\|\mathbf{v}-\mathbf{z}\right\|_2^2 = \underset{\{\mathbf{z}_2 \in \mathbb{R}_+\}}{\arg\min}\, \left\|\mathbf{z}_2 - \mathbf{v}\right\|_2^2 = \Pi_{\mathbb{R}_+}\left(\mathbf{v}\right), \tag{47}$$

where $\Pi_{\mathbb{R}_+}\left(\cdot\right)$ is the element-wise projection operator onto the convex set $\mathbb{R}_+$ (the nonnegativity constraints)

$$\Pi_{\mathbb{R}_+}\left(\mathbf{v}_j\right) = \begin{cases} 0 & \mathbf{v}_j < 0 \\ \mathbf{v}_j & \mathbf{v}_j \geq 0 \end{cases} \tag{48}$$

For more details on this and other proximal operators, please refer to [Boyd et al. 2011; Parikh and Boyd 2014].

3.4 Poisson Deconvolution with ADMM and TV

In many applications, it may be desirable to add additional priors to the Poisson deconvolution. Expressions for TV-regularized RL updates have been derived, but these expressions are not very flexible w.r.t. changing the prior. Further, it may be challenging to derive multiplicative update rules that guarantee non-negativity throughout the iterations in the same fashion as RL does.

ADMM can be used to add arbitrary (convex) priors to the denoising process, we derive expressions for total variation in this section. For this purpose, we extend Equation 32 by two additional penalty functions, such that

$$g_1\left(\mathbf{z}_1\right) = -\log\left(p\left(\mathbf{b}|\mathbf{z}_1\right)\right), \quad g_2\left(\mathbf{z}_2\right) = \mathcal{I}_{\mathbb{R}_+}\left(\mathbf{z}_2\right), \quad g_3\left(\mathbf{z}_3\right) = \lambda\left\|\mathbf{z}_3\right\|_1, \quad g_4\left(\mathbf{z}_4\right) = \lambda\left\|\mathbf{z}_4\right\|_1 \tag{49}$$

Our TV-regularized problem written in ADMM standard form then becomes

$$\begin{aligned} \underset{\{\mathbf{x}\}}{\text{minimize}} \quad & -\log\left(p\left(\mathbf{b}|\mathbf{z}_1\right)\right) + \mathcal{I}_{\mathbb{R}_+}\left(\mathbf{z}_2\right) + \lambda\left\|\mathbf{z}_3\right\|_1 + \lambda\left\|\mathbf{z}_4\right\|_1 \\ \text{subject to} \quad & \underbrace{\begin{bmatrix} \mathbf{A} \\ \mathbf{I} \\ \mathbf{D}_x \\ \mathbf{D}_y \end{bmatrix}}_{\mathbf{K}} \mathbf{x} - \underbrace{\begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \\ \mathbf{z}_3 \\ \mathbf{z}_4 \end{bmatrix}}_{\mathbf{z}} = 0 \end{aligned} \tag{50}$$

The Augmented Lagrangian of this objective is formulated as

$$L_\rho\left(\mathbf{x},\mathbf{z},\mathbf{y}\right) = \sum_{i=1}^{4} g_i\left(\mathbf{z}_i\right) + \mathbf{y}^T\left(\mathbf{K}\mathbf{x}-\mathbf{z}\right) + \frac{\rho}{2}\left\|\mathbf{K}\mathbf{x}-\mathbf{z}\right\|_2^2, \tag{51}$$

which leads to the following iterative updates

for $k = 1$ **to** $maxiter$

$$\mathbf{x} \leftarrow \mathbf{prox}_{quad}\left(\mathbf{v}\right) = \underset{\{\mathbf{x}\}}{\arg\min}\, L_\rho\left(\mathbf{x},\mathbf{z},\mathbf{y}\right) = \underset{\{\mathbf{x}\}}{\arg\min}\, \frac{1}{2}\left\|\mathbf{K}\mathbf{x}-\mathbf{v}\right\|_2^2, \quad \mathbf{v} = \mathbf{z}-\mathbf{u} \tag{52}$$

$$\mathbf{z}_1 \leftarrow \mathbf{prox}_{\mathcal{P},\rho}\left(\mathbf{v}\right), \quad \mathbf{v} = \mathbf{A}\mathbf{x}+\mathbf{u}_1 \tag{53}$$

$$\mathbf{z}_2 \leftarrow \mathbf{prox}_{\mathcal{I},\rho}\left(\mathbf{v}\right), \quad \mathbf{v} = \mathbf{x}+\mathbf{u}_2 \tag{54}$$

$$\mathbf{z}_3 \leftarrow \mathbf{prox}_{\|\cdot\|_1,\rho}\left(\mathbf{v}\right), \quad \mathbf{v} = \mathbf{D}_x\mathbf{x}+\mathbf{u}_3 \tag{55}$$

$$\mathbf{z}_4 \leftarrow \mathbf{prox}_{\|\cdot\|_1,\rho}\left(\mathbf{v}\right), \quad \mathbf{v} = \mathbf{D}_y\mathbf{x}+\mathbf{u}_4 \tag{56}$$

$$\underbrace{\begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \mathbf{u}_3 \\ \mathbf{u}_4 \end{bmatrix}}_{\mathbf{u}} \leftarrow \mathbf{u}+\mathbf{K}\mathbf{x}-\mathbf{z}$$

end for

The proximal operator of the Poisson denoiser (Eq. 53) and for the indicator function (Eq. 54) are already derived in Sections 3.3.2 and 3.3.3, respective. These proximal operators do not change for the TV-regularized version of the Poisson deconvolution problem.

The proximal operators for the quadratic problem (Eq. 52) and for the ℓ_1-norm (Eqs. 55 and 56) are derived in the following.

3.4.1 Proximal Operator for Quadratic Subproblem $\mathbf{prox}_{quad}$

For the proximal operator of the quadratic subproblem (Eq. 52), we formulation the closed-form solution via the normal equations. This is similar to Section 3.3.1 but extended by the gradient matrices:

$$\begin{aligned}\mathbf{prox}_{quad}(\mathbf{v}) &= \underset{\{\mathbf{x}\}}{\arg\min}\ \frac{1}{2}\left\| \underbrace{\begin{bmatrix}\mathbf{A}\\ \mathbf{I}\\ \mathbf{D}_x\\ \mathbf{D}_y\end{bmatrix}}_{\mathbf{K}}\mathbf{x} - \underbrace{\begin{bmatrix}\mathbf{v}_1\\ \mathbf{v}_2\\ \mathbf{v}_3\\ \mathbf{v}_4\end{bmatrix}}_{\mathbf{v}}\right\|_2^2 = \left(\mathbf{K}^T\mathbf{K}\right)^{-1}\mathbf{K}^T\mathbf{v} \\ &= \left(\mathbf{A}^T\mathbf{A} + \mathbf{I} + \mathbf{D}_x^T\mathbf{D}_x + \mathbf{D}_y^T\mathbf{D}_y\right)^{-1}\left(\mathbf{A}^T\mathbf{v}_1 + \mathbf{v}_2 + \mathbf{D}_x^T\mathbf{v}_3 + \mathbf{D}_y^T\mathbf{v}_4\right) \\ &= \mathcal{F}^{-1}\left(\frac{\mathcal{F}\{c\}^*\cdot\mathcal{F}\{v_1\} + \mathcal{F}\{v_2\} + \mathcal{F}\{d_x\}^*\cdot\mathcal{F}\{v_3\} + \mathcal{F}\{d_y\}^*\cdot\mathcal{F}\{v_4\}}{\mathcal{F}\{c\}^*\cdot\mathcal{F}\{c\} + 1 + \mathcal{F}\{d_x\}^*\cdot\mathcal{F}\{d_x\} + \mathcal{F}\{d_y\}^*\cdot\mathcal{F}\{d_y\}}\right)\end{aligned} \tag{57}$$

Note that this proximal operator does not require explicit matrix-vector products to be computed. The structure of this specific problem allows for a closed-form solution to be expressed using a few Fourier transforms as well as element-wise multiplications and divisions. Typically, $\mathcal{F}\{c\}^*, \mathcal{F}\{d_x\}^*, \mathcal{F}\{d_y\}^*$ and the denominator can be precomputed and do not have to be updated throughout the ADMM iterations.

3.4.2 Proximal Operator for ℓ_1-norm $\mathbf{prox}_{\|\cdot\|_1,\rho}$

Remember from the notes of lecture 6 that the ℓ_1-norm is convex but not differentiable. A closed form solution for the proximal operator exists, such that

$$\mathbf{prox}_{\|\cdot\|_1,\rho}(\mathbf{v}) = \mathcal{S}_{\lambda/\rho}(\mathbf{v}) = \underset{\{\mathbf{z}\}}{\arg\min}\ \lambda\|\mathbf{z}\|_1 + \frac{\rho}{2}\|\mathbf{v}-\mathbf{z}\|_2^2, \tag{58}$$

with $\mathcal{S}_\kappa(\cdot)$ being the element-wise soft thresholding operator

$$\mathcal{S}_\kappa(v) = \begin{cases} v-\kappa & v > \kappa \\ 0 & |v| \le \kappa \\ v+\kappa & v < -\kappa \end{cases} = (v-\kappa)_+ - (-v-\kappa) \tag{59}$$

3.5 Poissonian-Gaussian Noise Modeling

In many imaging applications, neither Gaussian-only nor Poisson-only noise is observed in the measured data, but a mixture of both. A detailed derivation of this model is beyond the scope of these notes, but a common approach (see e.g. [Foi et al. 2008]) to model Poissonian-Gaussian noise is to approximate the Poisson-term by a signal-dependent Gaussian distribution $\mathcal{N}(\mathbf{Ax}, \mathbf{Ax})$, such that the measurements follow a signal-dependent Gaussian distribution

$$\mathbf{b} \sim \mathcal{N}(\mathbf{Ax}, \mathbf{Ax}) + \mathcal{N}\left(0, \sigma^2\right) = \mathcal{N}(\mathbf{Ax}, 0) + \mathcal{N}(0, \mathbf{Ax}) + \mathcal{N}\left(0, \sigma^2\right) \tag{60}$$

$$= \mathbf{Ax} + \eta, \quad \eta \sim \mathcal{N}\left(0, \mathbf{Ax} + \sigma^2\right) \tag{61}$$

Therefore, the probability of the measurements is

$$p(\mathbf{b}|\mathbf{x}, \sigma) = \frac{1}{\sqrt{2\pi}\sqrt{\mathbf{Ax}+\sigma^2}} e^{-\frac{(\mathbf{b}-\mathbf{Ax})^2}{2(\mathbf{Ax}+\sigma^2)}}. \tag{62}$$

A proximal operator for this formulation can be derived similar to those discussed in the previous sections.

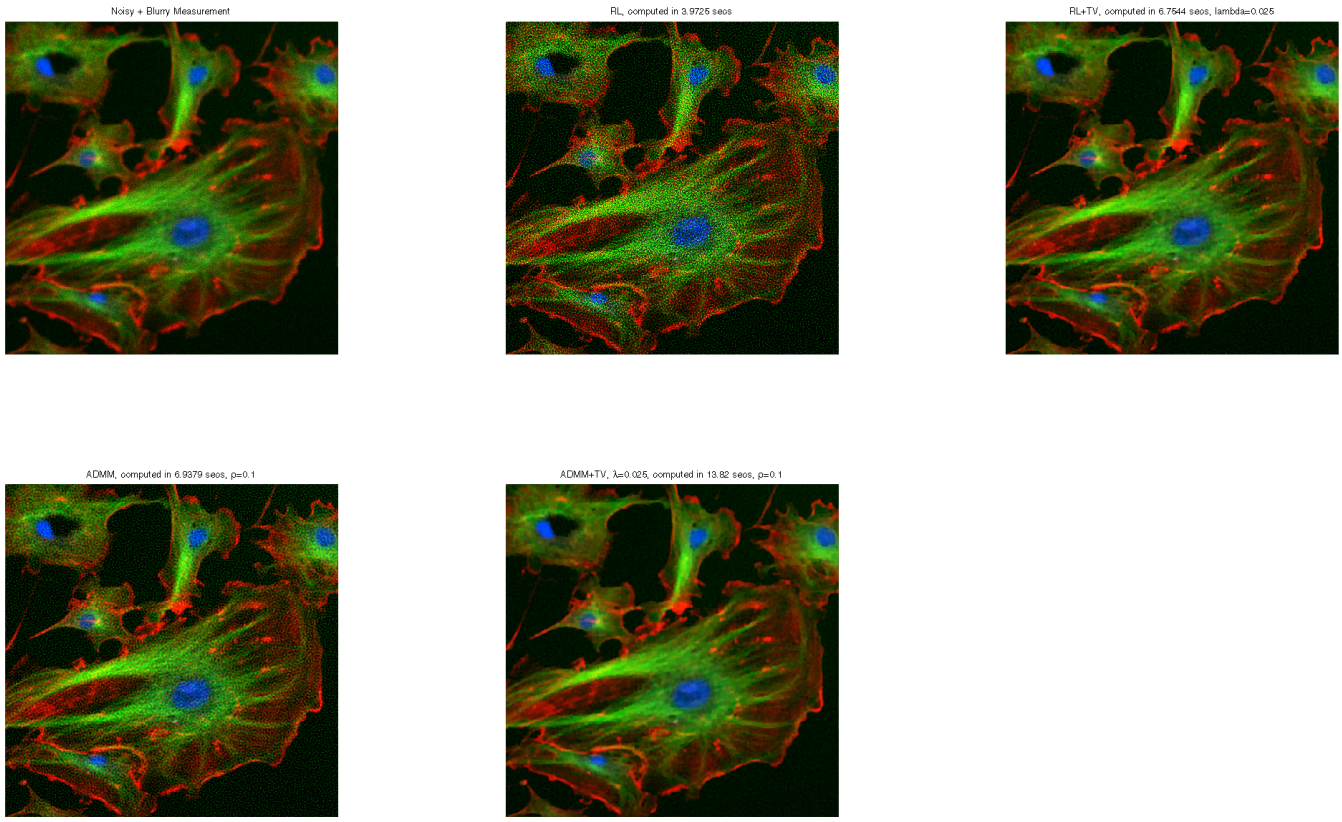

Figure 2: *Comparison of different image reconstruction techniques for a blurry image with Poisson noise (top left). The benchmark is Richardson-Lucy (top center), which results in a noisy reconstruction. Adding a TV-prior to the multiplicative update rules of RL enhances the reconstruction significantly (top, right). Here, we implement ADMM with proper non-negativity constraints and include the proximal operator for the MAP estimate under Poisson noise (bottom left). ADMM combined with a simple TV prior (bottom center) achieves slightly improved results compared to RL+TV but the quality is comparable.*

Note that Alessandro Foi provides a convenient Matlab library to calibrate the signal-dependent and signal-independent noise parameters of a Poissonian-Gaussian noise model from a single camera images on the following website: `http://www.cs.tut.fi/~foi/sensornoise.html`.

An alternative way to dealing with mixed Poisson-Gaussian noise is to approximate this using a *shifted Poisson distribution* [Snyder et al. 1993; Snyder et al. 1995; Chouzenoux et al. 2015; Chakrabarti and Zickler 2012; Ikoma et al. 2018]. This is a very accurate approximation that directly results in a closed-form solution for the proximal operator, as derived by Ikoma et al. [2018] and others.

References

BOYD, S., PARIKH, N., CHU, E., PELEATO, B., AND ECKSTEIN, J. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning 3*, 1, 1–122.

BUADES, A., COLL, B., AND MOREL, J. M. 2005. A non-local algorithm for image denoising. In *Proc. IEEE CVPR*, vol. 2, 60–65.

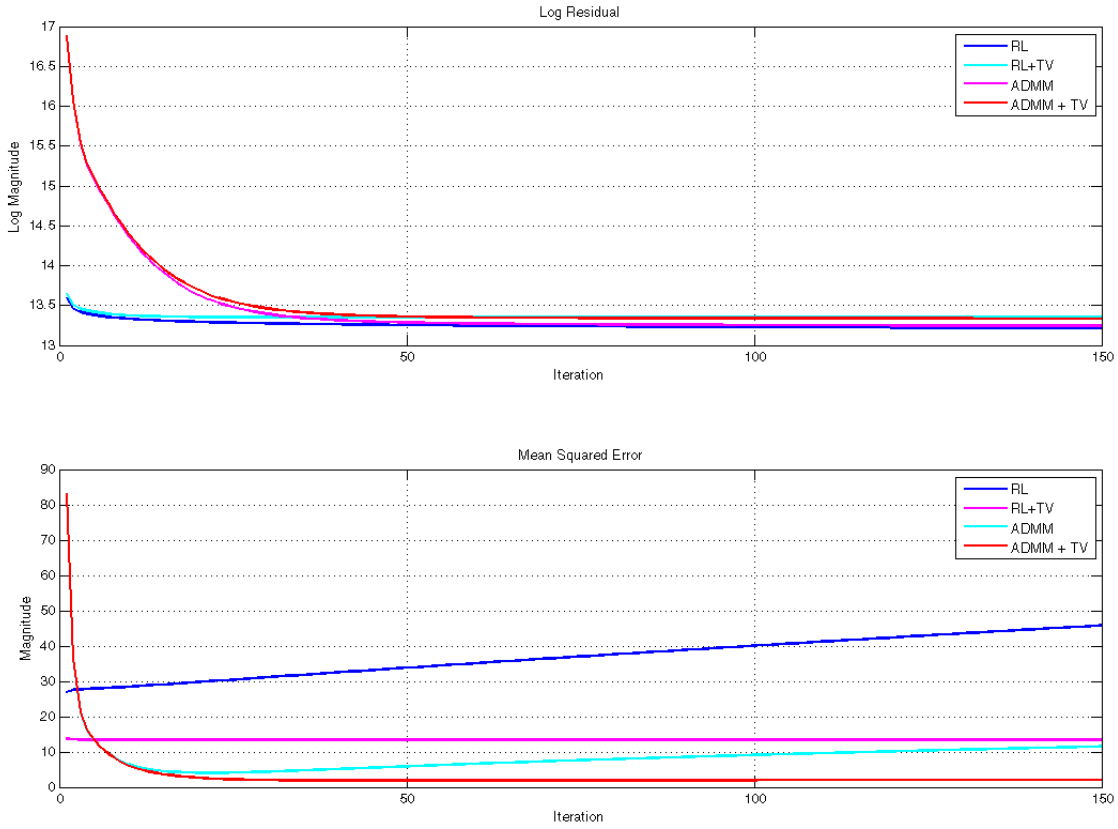

Figure 3: *Convergence and mean squared error for the cell example. Although RL and RL+TV converge to a low residual quickly, the MSE of RL actually blows up after a few iterations (noisy is significantly amplified). RL+TV prevents that, but ADMM+TV finds a better solution than RL+TV, as seen in the MSE plots.*

CHAKRABARTI, A., AND ZICKLER, T. 2012. Image restoration with signal-dependent camera noise. In *arXiv 1204.2994*.

CHOUZENOUX, E., JEZIERSKA, A., PESQUET, J.-C., AND TALBOT, H. 2015. A convex approach for image restoration with exact poissongaussian likelihood. *SIAM J. Imaging Sci. 8*, 4, 2662–2682.

DABOV, K., FOI, A., KATKOVNIK, V., AND EGIAZARIAN, K. 2007. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on Image Processing 16*, 8, 2080–2095.

DEY, N., BLANC-FERAUD, L., ZIMMER, C., KAM, Z., OLIVO-MARIN, J.-C., AND ZERUBIA, J. 2004. 3d microscopy deconvolution using richardsonlucy algorithm with total variation regularization. *INRIA Technical Report*.

DEY, N., BLANC-FERAUD, L., ZIMMER, C., KAM, Z., OLIVO-MARIN, J.-C., AND ZERUBIA, J. 2004. A deconvolution method for confocal microscopy with total variation regularization. In *IEEE Symposium on Biomedical Imaging: Nano to Macro*, 1223–1226.

DUPE, F.-X., FADILI, M., AND STARCK, J.-L. 2011. Inverse problems with poisson noise: Primal and primal-dual splitting. In *Proc. IEEE International Conference on Image Processing (ICIP)*, 1901–1904.

FIGUEIREDO, M., AND BIOUCAS-DIAS, J. 2010. Restoration of poissonian images using alternating direction optimization. *IEEE Transactions on Image Processing 19*, 12, 3133–3145.

FOI, A., TRIMECHE, M., KATKOVNIK, V., AND EGIAZARIAN, K. 2008. Practical poissonian-gaussian noise modeling and fitting for single-image raw-data. *IEEE Trans. Image Processing 17*, 10, 1737–1754.

IKOMA, H., BROXTON, M., KUDO, T., AND WETZSTEIN, G. 2018. A convex 3d deconvolution algorithm for low photon count fluorescence imaging. *Scientific Reports 8*, 11489.

LUCY, L. B. 1974. An iterative technique for the rectification of observed distributions. *Astron. J. 79*, 745–754.

PARIKH, N., AND BOYD, S. 2014. Proximal algorithms. *Foundations and Trends in Machine Learning 1*, 3, 123–231.

RICHARDSON, W. H. 1972. Bayesian-based iterative method of image restoration$*$. *J. Opt. Soc. Am. 62*, 1, 55–59.

SNYDER, D. L., HAMMOUD, A. M., AND WHITE, R. L. 1993. Image recovery from data acquired with a charge-coupled-device camera. *J. Opt. Soc. Am. A 10*, 5, 10141023.

SNYDER, D. L., HELSTROM, C. W., LANTERMAN, A., FAISAL, M., AND WHITE, R. L. 1995. Compensation for readout noise in ccd images. *J. Opt. Soc. Am. A 12*, 2, 272283.

WEN, Y., CHAN, R. H., AND ZENG, T. 2016. Primal-dual algorithms for total variation based image restoration under poisson noise. *Science China Mathematics 59*, 1, 141–160.