# Analysis of Richardson-Lucy Deconvolution with a TV Prior

Jerald R. Evans

*EE367 Project Report*

(Dated: March 19, 2021)

ABSTRACT

This report describes an analysis of Richardson-Lucy deconvolution using a total variation prior based on two different update rules, derived from an incorrect and a correct expression for the gradient of the L1 norm of the x and y gradients. Qualitative and quantitative comparisons of the two algorithms are shown.

## I. INTRODUCTION

The Richardson-Lucy method [1] [2] can be used to reconstruct an image which has been acquired in a process that introduces Poisson noise. The log-likelihood equation is derived from a joint probability equation based on independent measurements involving only Poisson noise [3]. Wetzstein [3] describes how to add a total variation prior to the Richardson-Lucy (RL) formulation. This will produce reconstructions with much lower mean squared error when compared to ground truth images. In Richardson-Lucy plus total variation (RL+TV), the log-likelihood expression includes a term which is a weighting factor ($\lambda$) times the L1 norms of the x and y gradients of the image summed together (an anisotropic TV norm). The gradient of this L1 norm must be taken in order to derive the RL+TV update rule for each iteration through the algorithm.

An earlier version of Wetzstein [3] (downloaded February 19, 2021) included the following equation for the gradient of the L1 norm:

$$\nabla\lambda\|\mathbf{D}\mathbf{x}\|_1 = \lambda\nabla\sum_{ij}\left(\sqrt{(\nabla_x\mathbf{x})_{ij}^2} + \sqrt{(\nabla_y\ \mathbf{x})_{ij}^2}\right) = -\lambda\left(\frac{\mathbf{D}_x\mathbf{x}}{|\mathbf{D}_x\mathbf{x}|} + \frac{\mathbf{D}_y\mathbf{x}}{|\mathbf{D}_y\mathbf{x}|}\right) \tag{1}$$

Here $\mathbf{D}_x$, for example, is the forward difference operator in x.

However, this expression is incorrect, as can be demonstrated in the 1D case:

$$\|\mathbf{D}\mathbf{x}\|_1 = \sum_i |\mathbf{x}_{i+1} - \mathbf{x}_i| \tag{2}$$

$$\left(\nabla_x\|\mathbf{D}\mathbf{x}\|_1\right)_i = \frac{\mathbf{x}_i - \mathbf{x}_{i-1}}{|\mathbf{x}_i - \mathbf{x}_{i-1}|} - \frac{\mathbf{x}_{i+1} - \mathbf{x}_i}{|\mathbf{x}_{i+1} - \mathbf{x}_i|} \tag{3}$$

So the gradient at position $i$ has two terms involving $x_i$, unlike Eq. 1.

An updated version of Wetzstein [3] (downloaded February 24, 2021) includes the following correct equation for the gradient of the L1 norm in the 2D case:

$$\nabla\lambda\|\mathbf{Dx}\|_1 = \lambda\nabla\sum_{ij}\left(\sqrt{(\nabla_x\mathbf{x})_{ij}^2} + \sqrt{(\nabla_y\ \mathbf{x})_{ij}^2}\right) = \lambda\left(\mathbf{D}_x^T\frac{\mathbf{D}_x\mathbf{x}}{|\mathbf{D}_x\mathbf{x}|} + \mathbf{D}_y^T\frac{\mathbf{D}_y\mathbf{x}}{|\mathbf{D}_y\mathbf{x}|}\right) \quad (4)$$

The two above expressions (Eqs. 1 and 4) lead to two different update rules for the RL+TV algorithm which are shown below. The purpose of this report is to analyze the effect on deconvolution results of using either the incorrect or correct update rule.

## II. RELATED WORK

Dey et al. [4] [5] give the following expression for the derivative of the L1 norm regularization term w.r.t. o:

$$\frac{\partial}{\partial o}J_{reg} = -\lambda div\left(\frac{\nabla o}{|\nabla o|}\right) \quad (5)$$

where *div* is the divergence. Dey et al. [4] note that in their notation $\mathbf{D}^T = -div$. This is consistent with Eq. 4 above (from the updated version of Wetzstein lecture notes [3]).

## III. METHODS

As noted above, the two expressions for the gradient of the L1 norm lead to two versions of the update rule for the RL+TV algorithm. We will refer to them as RL+TV_A and RL+TV_B, where A refers to the incorrect update rule and B refers to the correct update rule.

RL+TV_A:

$$\mathbf{x}^{(q+1)} = \frac{\mathbf{A}^T(\frac{\mathbf{b}}{\mathbf{Ax}})}{\mathbf{A}^T\mathbf{1} - \lambda\left(\frac{\mathbf{D}_x\mathbf{x}}{|\mathbf{D}_x\mathbf{x}|} + \frac{\mathbf{D}_y\mathbf{x}}{|\mathbf{D}_y\mathbf{x}|}\right)} \cdot \mathbf{x}^{(q)} \quad (6)$$

RL+TV_B:

$$\mathbf{x}^{(q+1)} = \frac{\mathbf{A}^T(\frac{\mathbf{b}}{\mathbf{Ax}})}{\mathbf{A}^T\mathbf{1} + \lambda\left(\mathbf{D}_x^T\frac{\mathbf{D}_x\mathbf{x}}{|\mathbf{D}_x\mathbf{x}|} + \mathbf{D}_y^T\frac{\mathbf{D}_y\mathbf{x}}{|\mathbf{D}_y\mathbf{x}|}\right)} \cdot \mathbf{x}^{(q)} \quad (7)$$

Here $\mathbf{A}$ represents a blur kernal, $\mathbf{x}$ represents the reconstructed image, and $\mathbf{b}$ represents a blurred and noisy image.

For this project, we implemented RL+TV deconvolution for several images and several values of $\lambda$.

For each image, we blur the image with a low-pass Gaussian kernel and then add Poisson noise. Then we run the two deconvolution algorithms on the blurry and noisy images.

We compare the results of the deconvolution using three measures of the sparsity of resulting image gradients (which is the goal of using the TV norm) and by calculating the log-likelihood and mean squared error for each result. For qualitative evaluation, we produce images of the gradient magnitudes of the resulting images.

To measure image gradient sparsity we of course calculate the L1 norm of the x and y image gradients, since this is the basis of the derivation of both of the update rules. In addition, we calculate the standard deviation of the histogram of the image gradients (explained in more depth below). The third measure is to define an arbitrary threshold of gradient magnitude and calculate the percentage of pixels in resulting images that fall below that threshold. The threshold we chose is one-half of the standard deviation of the gradient magnitude of the blurred, noisy image that we use as input to the deconvolution algorithms.

Measuring the standard deviation of the gradient histogram is based on the following observation. The gradient histogram is calculated by flattening the gradient arrays for both x and y and all color channels for all pixels into a 1D array. If we look at such histograms for a single ground truth image and the corresponding blurred and noisy image, we see something like **Figure 1**. Here we are using the FluorescentCells.jpg image that was used in Homework 6 for this course.

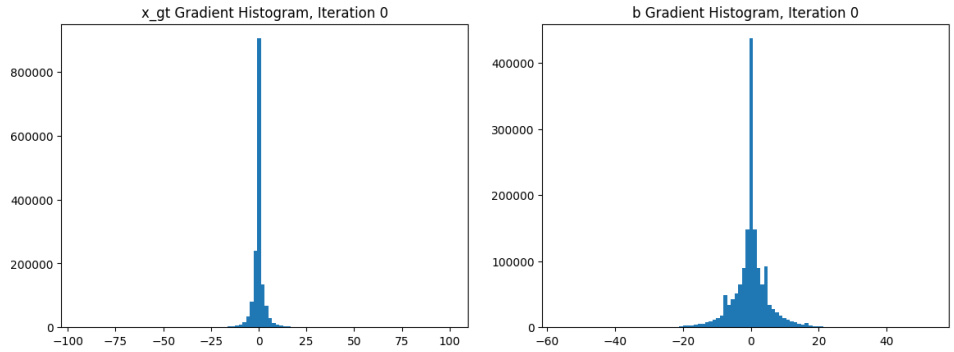


FIG. 1: Gradient histograms of ground truth & blurred, noisy images

Note that the histogram of the ground truth image is tall and skinny while that of the blurred and noisy image is shorter and fatter. We would expect that over the course of iterations of the RL+TV algorithm the histogram of the resulting image would become more like the ground truth histogram. We track this by computing the standard deviation of the result at each iteration.

We also use a more heuristic method to try to get an intuitive feel for the differences in the two algorithms. We compare a known good result for RL+TV_A to the result

for RL+TV_B operating on the same image with the same value of $\lambda$. Since the B result is not as good as the A result, we attempt to determine a value of $\lambda$ to use with RL+TV_B to produce a comparable good result. We do this by doing a naive analysis of the denominators in the two update rules (again using histograms). The details are discussed below.

## IV. HEURISTIC ANALYSIS & EVALUATION OF RL+TV_A & RL+TV_B

In Homework 6 we implemented RL+TV_A for a range of $\lambda$ values and found that $\lambda = 0.04$ worked well to reconstruct the blurred and noisy FluorescentCells.jpg image. We started the analysis of the two RL+TV algorithms by comparing them at $\lambda = 0.04$. The original image and the gradient magnitude results are shown in **Figure 2**.

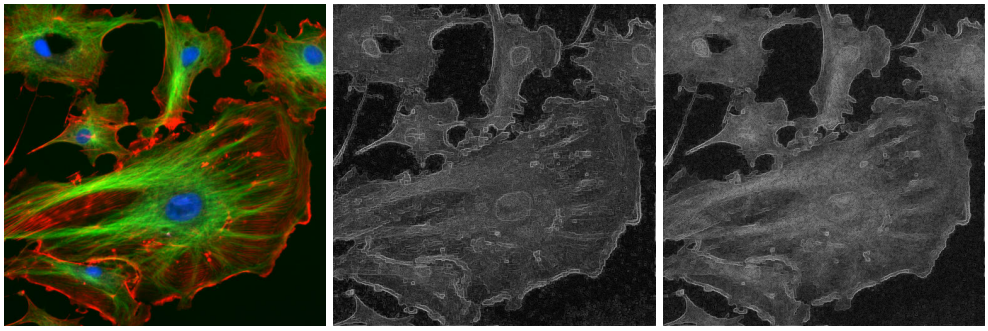

FIG. 2: Ground truth image and gradient magnitude images for RL+TV_A and RL+TV_B after 100 iterations at $\lambda = 0.04$

At $\lambda = 0.04$, RL+TV_A produces a better gradient magnitude image than RL+TV_B, showing sharper edges. In order to better understand if we can adjust $\lambda$ in RL+TV_B to get a result as good as RL+TV_A, we take a closer look at the corresponding update rules, specifically the denominators of those rules where the different expressions for the gradient of the L1 norm appear. See **Figure 3**.

$$\mathbf{A}^T\mathbf{1} - \lambda\left(\frac{\mathbf{D}_x\mathbf{x}}{|\mathbf{D}_x\mathbf{x}|} + \frac{\mathbf{D}_y\mathbf{x}}{|\mathbf{D}_y\mathbf{x}|}\right) \qquad \mathbf{A}^T\mathbf{1} + \lambda\left(\mathbf{D}_x^T\frac{\mathbf{D}_x\mathbf{x}}{|\mathbf{D}_x\mathbf{x}|} + \mathbf{D}_y^T\frac{\mathbf{D}_y\mathbf{x}}{|\mathbf{D}_y\mathbf{x}|}\right)$$

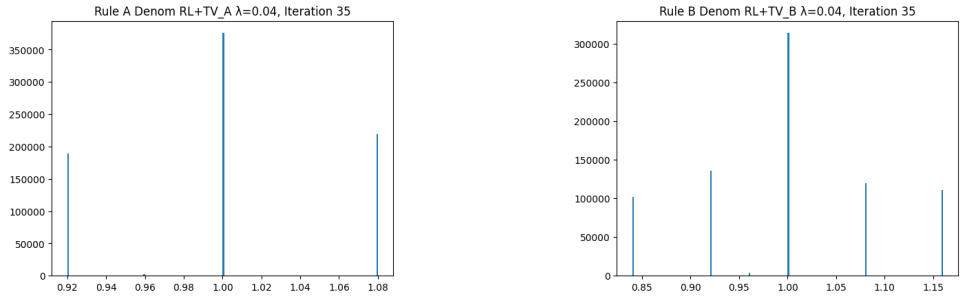


FIG. 3: Histograms of the denominators of the update rules

For the denominator of the RL+TV_A update rule, we see that the quantity in parentheses takes integer values ranging from -2 to 2 (per pixel), while the corresponding quantity for RL+TV_B takes integer values ranging from -4 to 4 (per pixel). If the quantity $\mathbf{A}^T\mathbf{1}$ is simple enough, we should see some structure in the total denominators. We take a simple approach and plot the histogram of the denominators for a particular iteration. For the Gaussian blur kernel we are using and this cells image, we get the results in **Figure 3**. We notice that the distribution width for RL+TV_B is twice that for RL+TV_A. Of course we can't make the distributions match exactly, and even if we could, the values at a particular pixel would not necessarily match, but the shape of the distributions suggests that running RL+TV_B at $\lambda = 0.02$ might produce a result as good as RL+TV_A at $\lambda = 0.04$. We show a quantitative comparison of these parameter values in **Figure 4**.

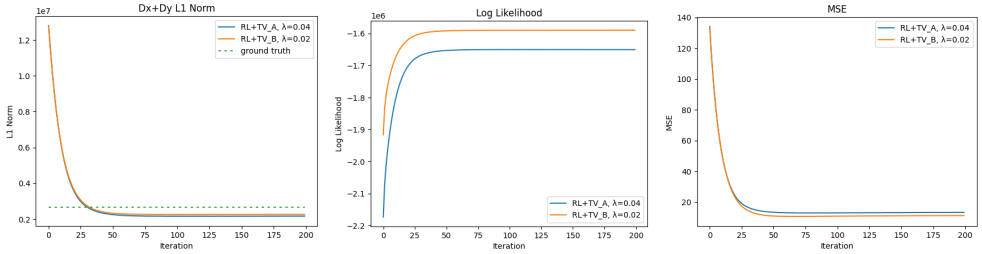


FIG. 4: Comparison of RL+TV_A, $\lambda = 0.04$ and RL+TV_B, $\lambda = 0.02$

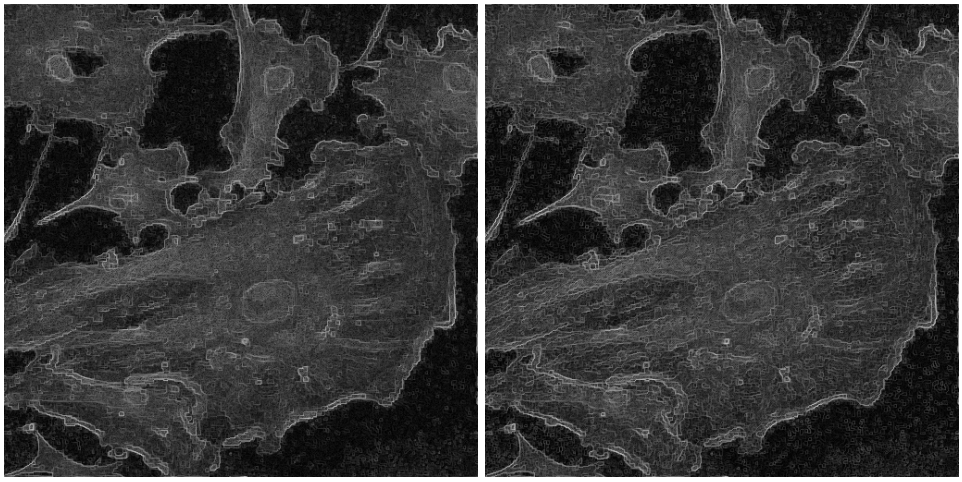

FIG. 5: Gradient magnitude after 200 iterations. Left- RL+TV_A, $\lambda = 0.04$; Right- RL+TV_B, $\lambda = 0.02$

We see that RL+TV_B now performs about as well as RL+TV_A in reducing the L1 norm of the image gradients, and produces higher log-likelihood and lower mean squared error. We also note that both algorithms have reduced the L1 norm below that of the ground truth image. In **Figure 5** we show the gradient magnitudes of the results for the two algorithms.

The main lesson to be drawn from this heuristic analysis is that the performance of these algorithms is highly sensitive to the choice of $\lambda$. It would also depend on the particular image being used and on the choice of blur kernel. In the next section we show results for various values of $\lambda$ and discuss results for different images. We have not analyzed different blur kernels.

## V. RESULTS

The FluorescentCells.jpg image was used to quantitatively compare RL+TV_A and RL+TV_B for $\lambda = 0.01, 0.02, 0.03, 0.04$ and $0.05$. In addition, RL without a prior was tested by setting $\lambda$=0.0. The three sparsity measures were calculated as well as log-likelihood and mean squared error for each parameter combination. The algorithms were run for 200 iterations. Results for RL+TV_A are shown in **Figure 6** and results for RL+TV_B are shown in **Figure 7**.

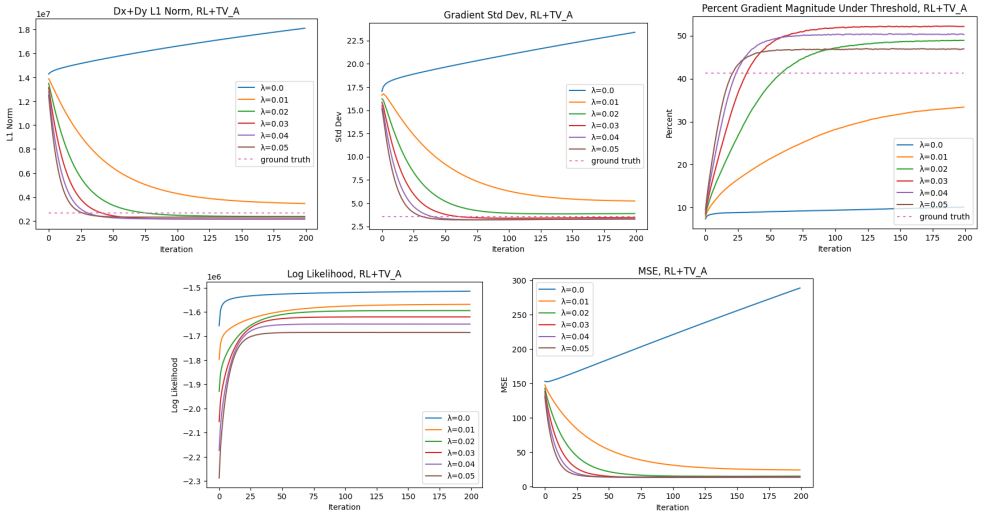


FIG. 6: RL+TV_A

In general we note that for this image and the range of $\lambda$ values tested, RL+TV_A performs best in reducing sparsity of the image gradients, while RL+TV_B produces higher log-likelihood and lower mean squared error.

The same algorithms and range of $\lambda$ values were tested using six other images: a second cell microscopy image, a Hubble telescope image, an image of a Mondrian painting (or imitation), a low-resolution television test pattern, a photograph of a coffee cup, and a photograph of a motorcycle. The results are not shown in this report, but the images and code to reproduce the data are included in the code submission for the project.

The above results generally hold for the additional images. For two images, the second microscopy image and the Mondrian image, RL+TV_A produces lower MSE than RL+TV_B at the optimum $\lambda$ for each algorithm for each image. The other results are consistent with those in **Figures 6** and **7**.

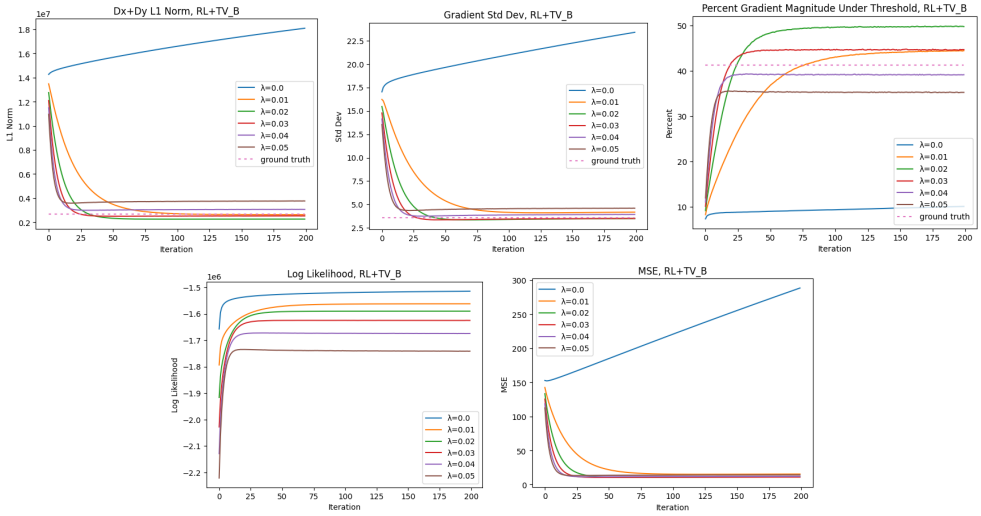


FIG. 7: RL+TV_B

For the two images for which RL+TV_A optimizes MSE, neither algorithm at any $\lambda$ reduces gradient sparsity below that of the ground truth image. Those are the only two images for which neither algorithm reduced sparsity below ground truth.

For each measure of sparsity or image quality, the optimum value of $\lambda$ varies depending on the image. For RL+TV_A optimum $\lambda$ ranges from 0.01 to 0.05, and the optimum is achieved at iteration numbers ranging from 90 to 199. For RL+TV_B optimum $\lambda$ ranges from 0.01 to 0.03 and the optimum is achieved at iteration numbers ranging from 41 to 199.

## VI. DISCUSSION

This project did not explore using different blur kernels. In addition, it is possible that finer gradations in tested $\lambda$ values could change the outlier results for RL+TV_A and MSE.

The fact that log-likelihood is always optimized with RL+TV_B is reassuring since the reason for taking the gradient of the L1 norm is to find an expression to maximize the log-likelihood including the TV prior.

Testing more microscopy images and more astronomical images would be desirable.

It would be interesting to test the algorithms using the central difference rather than the forward difference to approximate image gradients.

It would also be interesting, but perhaps not possible, to try to derive an analytic expression for the prior term that is being optimized by RL+TV_A.

The main conclusions from the project are:

1. RL+TV_B always produces higher log-likelihood
2. RL+TV_A always produces sparser gradients
3. RL+TV_B usually produces lower mean squared error

provided the optimal $\lambda$ is chosen (where optimal $\lambda$ depends on the image and the quantity being optimized).

---

## REFERENCES


[1] Richardson, W. H.. Bayesian-based iterative method of image restoration. J. Opt. Soc. Am. 62, 1, 55–59.

[2] Lucy, L. B. An iterative technique for the rectification of observed distributions. Astron. J. 79, 745–754.

[3] Wetzstein, G., EE367 / CS448I Computational Imaging and Display Notes: Noise, Denoising, and Image Reconstruction with Noise (lecture 10). http://stanford.edu/class/ee367/reading/lecture10_notes.pdf

[4] Dey, N., Blanc-Feraud, L., Zimmer, C., Kam, Z., Olivo-Marin, J.-C., and Zerubia, J. 3d microscopy deconvolution using Richardson-Lucy algorithm with total variation regularization. INRIA Technical Report.

[5] Dey, N., Blanc-Feraud, L., Zimmer, C., Kam, Z., Olivo-Marin, J.-C., and Zerubia, J. A deconvolution method for confocal microscopy with total variation regularization. In IEEE Symposium on Biomedical Imaging: Nano to Macro, 1223–1226.