Flutter Shutter Deconvolution Methods Under Gaussian and Poisson Noise

Ifueko Igbinedion
Stanford University
EE 367, Winter 2016
Final Project Report
ifueko@stanford.edu

Abstract

The flutter shutter camera capture method developed by Raskar et al. [4] preserves high spatial frequencies of moving objects by "fluttering" the camera's aperture open and closed during the exposure time. This form of exposure coding changes the camera's filter to a broadband filter, preserving many high spatial frequencies that would have been lost with the traditional camera's box filter. This project explores the importance and impact of noise on various total variation regularized deconvolution methods that are well-posed due to flutter shutter capture. Using Gaussian and Poisson noise models, this project illustrates the various tradeoffs for utilizing different regularization terms in the presence of noise.

Introduction

Both motion blur and noise during image capture make image deconvolution a difficult task. While object motion causes blur in images captured with traditional cameras, if the direction and velocity of a moving object can be estimated, capture methods can be modified to make deconvolution of these motion blurred images more robust. Researchers proposed the flutter shutter camera [4], in which the cameras shutter is "fluttered" open and closed during the exposure time. While the circular point spread function of traditional cameras destroys high spatial frequencies of moving objects in an image, the flutter shutter camera's point spread function preserves these high frequencies, allowing them to be accurately reconstructed during deconvolution. Noise further pollutes these high spatial frequencies, and if not accounted for it can make restoring a good image an impossible task.

The first part of this paper defines general Gaussian and Poisson noise models for imaging applications, describes the specific image formation of a flutter shutter image, and discusses synthetic image generation utilizing the two concepts. The second part describes two variations of the total variation regularized alternating direction method of multipliers (TV-ADMM), the iterative optimization scheme used to deconvolve flutter-shutter generated images with varying amounts of noise. The last part of this paper discusses the experimental results, comparing the performance of the two methods, and explores future research and applications of flutter shutter deconvolution.

1. Image Formation

We consider scenes with moving objects polluted by varying degrees of noise. The general image formation model is

$$\mathbf{b} = \mathbf{A}\mathbf{x} + \eta,$$

where $\mathbf{b} \in \Re^M$ is a vector of measurements, $\mathbf{x} \in \Re^N$ is a vector of unknown pixel values, $\mathbf{A} \in \Re^{M \times N}$ is the smearing matrix of the system, and η is an additive noise term. In this project, the smearing matrix $\mathbf{A}$ is a direct result of the fluttering code used by our capture system, and the additive noise term η is Gaussian i.i.d and/or signal-dependent Poisson noise. In the next section, the models for Gaussian only, Poisson only, and mixed Gaussian-Poisson noise are described.

1.1. Noise Modeling

In many imaging applications, collected image measurements are often corrupted by noise. This noise can come from a variety of different sources, including thermal, amplifier and read noise, which follows a Gaussian distribution, and signal dependent shot noise caused by quantum fluctuations of traveling light as well as photon to electron conversion, which follows a Poisson distribution [7].

When the noise term η follows a zero mean i.i.d., Gaussian only distribution, it is modeled as $\eta \sim \mathcal{N}(0, \sigma^2)$ The noise free signal is modeled as $\mathbf{A}\mathbf{x} \sim \mathcal{N}(\mathbf{A}\mathbf{x}, 0)$. Therefore, the noisy image $\mathbf{b}$ can be modeled as a gaussian distribution, $\mathbf{b} \sim \mathcal{N}(\mathbf{A}\mathbf{x}, \sigma^2)$, which follows a per-pixel

normal distribution

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

[7].

In applications which shot noise dominates, image formation is modeled by the Poisson process, $\mathbf{b} \sim \mathcal{P}(\mathbf{Ax})$. However, in many imaging applications it is common exploit the normal approximation of the Poisson distribution, $\mathcal{P}(\lambda) \approx \mathcal{N}(\lambda, \lambda)$[2] [7], such that the both the noise term η and measured image $\mathbf{b}$ follow a normal distributions, $\eta \sim \mathcal{N}(0, \mathbf{Ax})$ and $\mathbf{b} \sim \mathcal{N}(\mathbf{Ax}, \mathbf{Ax})$. This follows a per pixel normal distribution

$$p(\mathbf{b_i}|\mathbf{x}) = \frac{(\mathbf{Ax})_i^{b_i} e^{-(\mathbf{Ax})}}{b_i!}$$

Thus, because the sum of two Gaussian distributions $y_1 \sim \mathcal{N}(\mu_1, \sigma_1)$ and $y_2 \sim \mathcal{N}(\mu_2, \sigma_2)$ is $y_1 + y_2 \sim \mathcal{N}(\mu_1 + \mu_2, \sigma_1 + \sigma_2)$, for mixed Gaussian-Poisson noise, the noise term follows a mixed normal distribution, $\eta \sim \mathcal{N}(0, \mathbf{Ax} + \sigma^2)$. Therefore our measured image follows a normal distribution, $\mathbf{b} \sim \mathcal{N}(\mathbf{Ax}, \mathbf{Ax} + \sigma^2)$, which follows a per-pixel normal distribution

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

[7]. These three noise models are utilized during synthetic image generation in this project.

1.2. Flutter Shutter Capture

The flutter shutter camera [4] aims to make deconvolution a well-posed problem by coding the exposure using a binary pseudo-random sequence c. Using this method, the motion of the object is temporally sampled in order to preserve the maximum range of spatial frequencies. In this project, we consider constant-velocity, fixed distance 1-d motion of objects in an image. Given a finite exposure time, the sensor integration time is divided into a number of chops such that each chop is of equal length. Each chop is assigned a pseudo-random binary value which dictates whether or not sensor integration occurs during that chop. Because we assume constant-velocity 1-d motion, the resulting moving object in the image can be modeled as the convolution of the flutter code with unknown pixel values: $\mathbf{b} = c * \mathbf{x}$, or the multiplication of the image by a circulant smearing matrix $\mathbf{A}$, whose eigenvalues are the Fourier transform of the fluttering code c: $\mathbf{b} = \mathbf{Ax}$. In this project, we utilize the near-optimal fluttering code[1] proposed by Raksar et al. [4].

[1]The flutter code proposed by Raskar et al. is the following sequence: 1010000111000001010000110011110111010111001001100111

Figure 1 shows the code sequence of a flutter shutter camera using this code as well as the code sequence used by traditional cameras, and the magnitude of their Discrete Fourier Transforms. We can see that the DFT of the flutter code resembles a broadband filter frequency response, while the DFT of the traditional exposure resembles a moving average filter. Their DFTs clearly illustrate why using a fluttered shutter improves deconvolution. First, because there are very few zeros in the flutter shutter frequency response, the high spatial frequencies in the image are preserved. Additionally, image reconstruction requires inversion in the frequency domain. With many zeros in the frequency response of the traditional camera's filter, an inverse filter would contain infinite values, and the corresponding deconvolution problem is ill-posed. The flutter shutter DFT, however, is invertible, and so the deconvolution becomes well posed under this capture system.

1.3. Synthetic Image Generation

Synthetic images are generated by taking a stationary image and shifting it in one direction, integrating based on the flutter code. Gaussian-only noise is simulated with additive noise using standard deviation σ, applied to the noise free image. Poisson-only noise is simulated by applying a Poisson distribution to the image, limited by the simulated number of photons ranging from $n = 50$ to $n = 1000$. Mixed Poisson-Gaussian noise is simulated by first applying the Poisson distribution to the image, and then applying additive Gaussian noise to the resulting image. Figure 2 shows the original and flutter shutter simulated images used in this project.

2. Total Variation Regularized Deconvolution with ADMM

A general formulation for image reconstruction is

$$\underset{\{x\}}{\text{minimize}} \; \frac{1}{2}\|\mathbf{Ax} - \mathbf{b}\|_2^2 + \Gamma(x),$$

where $\mathbf{A}$, $\mathbf{x}$ and $\mathbf{b}$ are defined as above, while $\Gamma(x)$ is a regularization term. In this project, we utilize two total-variation [5] based regularization terms. The first, anisotropic term is modeled as

$$\Gamma_1(x) = \lambda\|\mathbf{Dx}\|_1$$

, where $\mathbf{D} = [\mathbf{D_x^T}\ \mathbf{D_y^T}]^T, \mathbf{D} \in \Re^{2MxN}$ is the finite differences approximation of the horizontal and vertical image gradients, as described in [6]. The second regularization term uses a weighted difference of isotropic and anisotropic terms, as proposed by Lou et al. [3], and is modeled as

$$\Gamma_2(x) = \|\mathbf{Dx}\|_1 - \lambda\|\mathbf{Dx}\|_2.$$

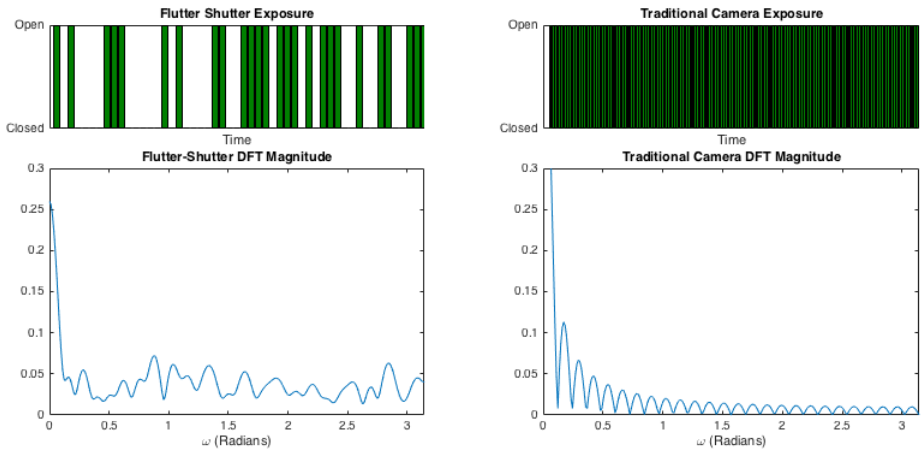
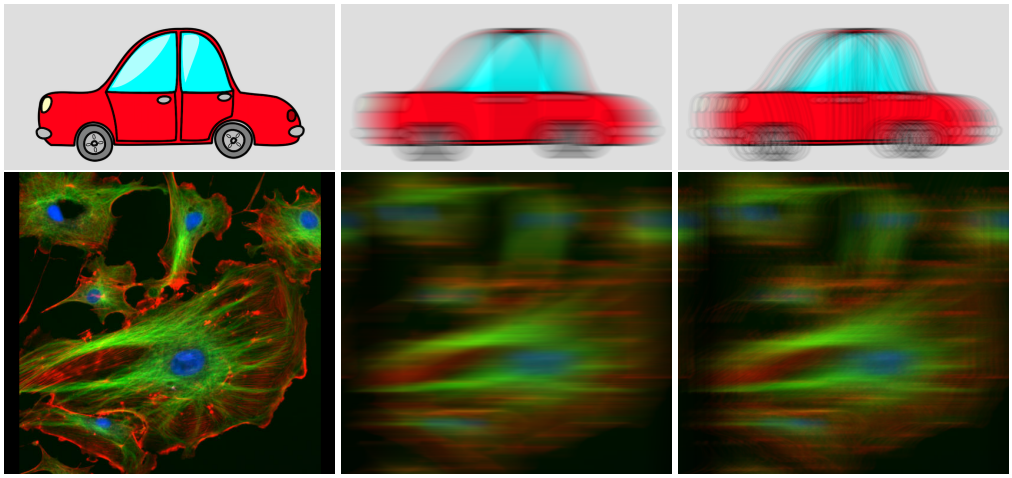


Figure 1: Flutter shutter vs. Traditional exposure. The DFT of the traditional camera contains many close to zero values at high frequencies, which destroys those frequencies. The fluttered shutter preserves the high spatial frequencies, as seen in its DFT.



(a) Stationary (b) Traditional Camera Motion Blurred (c) Flutter Shutter Camera Motion Blurred

Figure 2: Stationary, traditional camera motion blurred and flutter shutter motion blurred images. We can see that the flutter shutter camera preserves details of the moving objects that otherwise would have been lost with traditional camera capture.

To preform this optimization for both regularization terms, the alternating direction method of multipliers (ADMM) algorithm proposed by Boyd et al. [1] is utilized. ADMM splits the objective function into a weighted sum of the individual functions in the objective, linked only by constraints [6]. Following ADMM strategy, the Augmented Lagrangian for the optimization problem is defined as

$$L_\rho(\mathbf{x}, \mathbf{y}, \mathbf{z}) = f(\mathbf{x}) + g(\mathbf{x}) + y^T(\mathbf{D}\mathbf{x} - \mathbf{z}) + \frac{\rho}{2}\|\mathbf{D}\mathbf{x} - \mathbf{z}\|_2^2$$

For the first regularization term, two auxiliary variables, $\mathbf{u}$ and z are incorporated. Boyd et al. [1] derive the following update rules for deconvolution using the anisotropic regularization term:

$$\mathbf{x} \leftarrow \underset{\{\mathbf{x}\}}{\arg\min}\, f(\mathbf{x}) + \frac{\rho}{2}\|\mathbf{D}\mathbf{x} - \mathbf{v}\|_2^2, \mathbf{v} = \mathbf{z} - \mathbf{u}$$

$$\mathbf{z} \leftarrow \underset{\{\mathbf{z}\}}{\arg\min}\, g(\mathbf{z}) + \frac{\rho}{2}\|\mathbf{v} - \mathbf{z}\|_2^2, \mathbf{v} = \mathbf{D}\mathbf{x} + \mathbf{u}$$

$$\mathbf{u} \leftarrow \mathbf{u} + \mathbf{D}\mathbf{x} - \mathbf{z}$$

For the second regularization term, we use similar update rules as derived by Lou et al. [3]. In order to decouple the

l1 and l2 norms used in the regularization term, we define a third auxiliary variable, $\mathbf{q}$ that links the l2 norm to the z update. The new z and q updates are

$$\mathbf{z} \leftarrow \underset{\{\mathbf{z}\}}{\arg\min}\, g(\mathbf{z}) + \frac{\rho}{2}\|\mathbf{v} - \mathbf{z}\|_2^2, \mathbf{v} = \mathbf{D}\mathbf{x} + \mathbf{u} + \lambda\mathbf{q}$$

$$\mathbf{q} \leftarrow \mathbf{D}\mathbf{x}/\sqrt{|\mathbf{D}\mathbf{x}|^2}$$

The $\mathbf{x}$ and $\mathbf{z}$ updates are implemented as proximal operators, as discussed in [7] and [1].

3. Experimental Results and Analysis

Table 1 shows the peak signal-to-noise ratio (PSNR) results of using the two regularizers on the synthetically generated flutter shutter images show in Figure 2. Figures 3 and 4 show the reconstructed images corresponding to the PSNRs. As expected, the PSNR decreases as noise increases or as photons become more limited. Against the noiseless measurements, the $\Gamma_2(x)$ regularizer performs better, achieving both higher PSNR values and sharper details in the reconstructed images. However, as noise increases, the $\Gamma_1(x)$ regularizer does a better job at suppressing noise in the image, achieving higher PSNR values than the $\Gamma_2(x)$ regularizer. This is likely due to the fact that because the $\Gamma_2(x)$ regularizer relies more heavily on the gradients of the image to guide the iterative descent. Overcompensation for gradients due to noise seems to have amplified the noise everywhere in the image, though details of the reconstructed moving object are still sharp.

Another interesting result of this project is the robustness of the algorithm against Poisson noise, especially in the fluorescent cells image. Against varying Poisson noise in the cells image, the reconstruction always results in a PSNR within 2db of the noiseless measurements. This suggests that flutter shutter deconvolution in general works well against Poisson noise, and could possibly be useful in imaging applications that are dominated by Poisson noise, such as microscopy, scientific imaging, and other low-light photography applications.

Future Work

Optical flow for Motion Detection

The capture setup assumes constant velocity, 1-d motion. If we were to capture multiple fluttered frames, it would be possible to use optical flow estimation to determine which portions of the image are moving and their relative velocity. This would not only streamline the flow from flutter shutter capture to deconvolution, which requires user input to properly scale and align the image to fit our motion assumptions.

One issue with the current deconvolution scheme is that the foreground and background are not separated before deconvolution. While the backgrounds of the test images in this project are one color, for patterned backgrounds, aliasing of the stationary pattern occurs during deconvolution. Optical flow estimation would also allow more accurate detection of and compensation for still backgrounds during reconstruction.

Auto code generation

This project utilizes the near-optimal code found by Raskar et al. [4]. This code is long, so for slow moving objects whose blur only spans a few pixels, this code is may not be optimal. Future research can be done in auto generation of flutter codes due to the perceived motion in the scene, using optical flow or hardware sensors. If codes can be generated on-the-fly that lead to good deconvolution, the flutter shutter camera could be one step closer to being natively supported by commercial cameras.

Acknowledgements

The author would like to thank Professor Gordon Wetzstein for his guidance and feedback on this project.

References

[1] Boyd, Stephen, et al. "Distributed optimization and statistical learning via the alternating direction method of multipliers." Foundations and Trends in Machine Learning 3.1 (2011): 1-122.

[2] Foi, Alessandro, et al. "Practical Poissonian-Gaussian noise modeling and fitting for single-image raw-data." Image Processing, IEEE Transactions on 17.10 (2008): 1737-1754.

[3] Lou, Yifei, et al. "A weighted difference of anisotropic and isotropic total variation model for image processing." SIAM Journal on Imaging Sciences 8.3 (2015): 1798-1823.

[4] Raskar, Ramesh, Amit Agrawal, and Jack Tumblin. "Coded exposure photography: motion deblurring using fluttered shutter." ACM Transactions on Graphics (TOG) 25.3 (2006): 795-804.

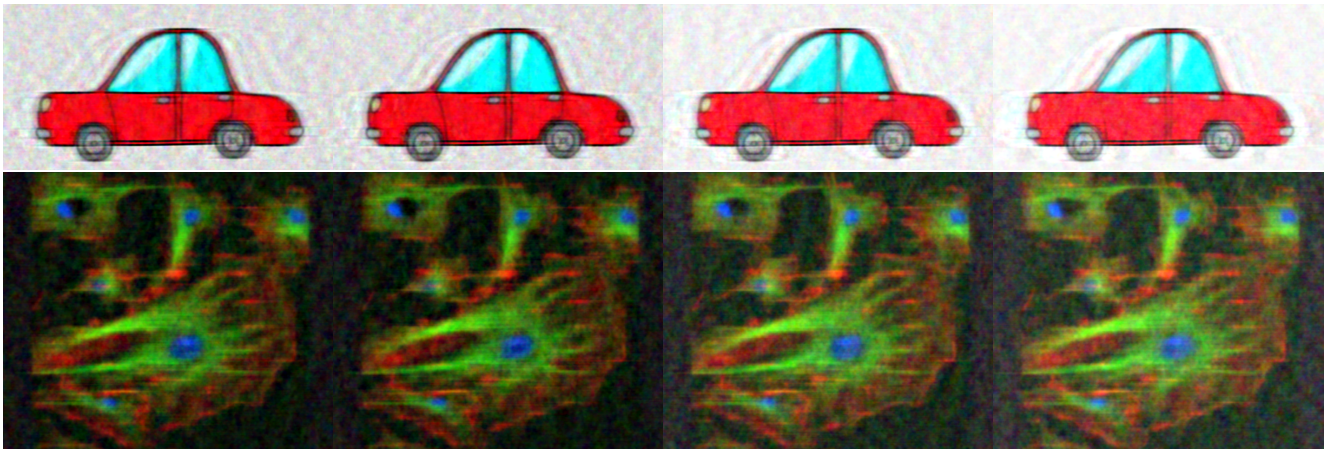
[5] Rudin, Leonid I., Stanley Osher, and Emad Fatemi. "Nonlinear total variation based noise removal algorithms." Physica D: Nonlinear Phenomena 60.1 (1992): 259-268.

[6] Wetzstein, Gordon. "Notes: Image Deconvolution (class 6)." Web.

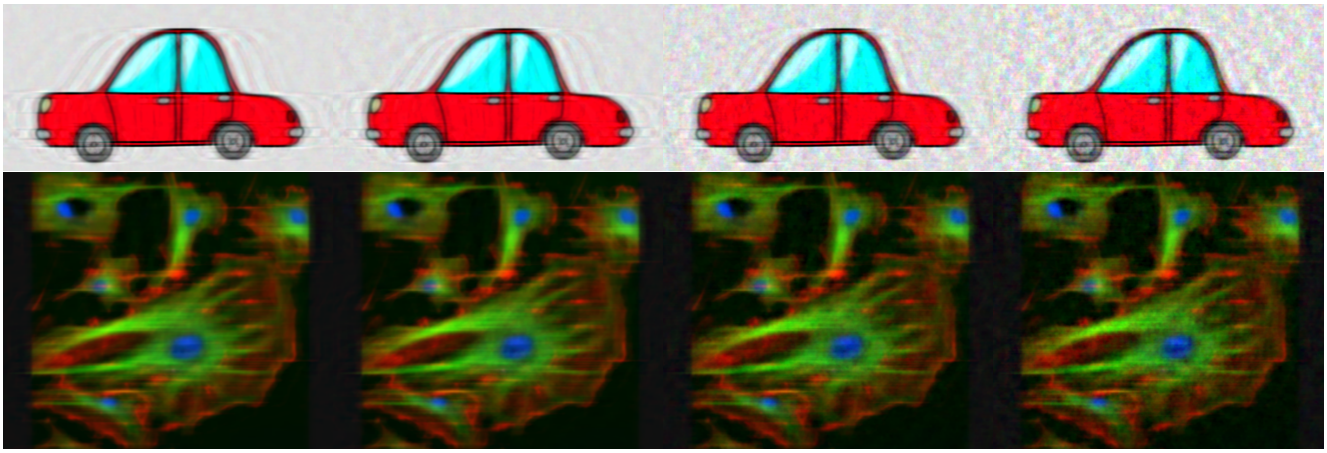
[7] Wetzstein, Gordon. "Notes: Noise and Image Reconstruction with Noise (lecture 11)." Web.

Regularizer	Noise parameters σ, n	PSNR: car.jpg	PSNR: cells.jpg
$\Gamma_1(x)$	Noiseless measurements	19.5655	25.5193
$\Gamma_1(x)$	Gaussian-only: $\sigma = 0.001$	19.1051	23.6385
$\Gamma_1(x)$	Gaussian-only: $\sigma = 0.01$	19.0911	23.4327
$\Gamma_1(x)$	Gaussian-only: $\sigma = 0.1$	17.5687	18.9986
$\Gamma_1(x)$	Gaussian-only: $\sigma = 0.15$	16.3378	16.4373
$\Gamma_1(x)$	Poisson-only: $n = 1000$ photons	19.5465	25.4887
$\Gamma_1(x)$	Poisson-only: $n = 500$ photons	19.5344	25.4687
$\Gamma_1(x)$	Poisson-only: $n = 100$ photons	19.4352	25.2549
$\Gamma_1(x)$	Poisson-only: $n = 50$ photons	19.3023	25.0222
$\Gamma_1(x)$	Gaussian+Poisson: $\sigma = 0.001$, $n = 1000$	19.1262	23.6505
$\Gamma_1(x)$	Gaussian+Poisson: $\sigma = 0.01$, $n = 500$	19.0358	23.3945
$\Gamma_1(x)$	Gaussian+Poisson: $\sigma = 0.001$, $n = 50$	18.8730	23.2928
$\Gamma_1(x)$	Gaussian+Poisson: $\sigma = 0.1$, $n = 100$	17.4967	18.8682
$\Gamma_2(x)$	Noiseless measurements	20.7761	26.0147
$\Gamma_2(x)$	Gaussian-only: $\sigma = 0.001$	18.9088	21.3250
$\Gamma_2(x)$	Gaussian-only: $\sigma = 0.01$	18.8828	21.1672
$\Gamma_2(x)$	Gaussian-only: $\sigma = 0.1$	17.0942	17.1554
$\Gamma_2(x)$	Gaussian-only: $\sigma = 0.15$	15.5478	14.9145
$\Gamma_2(x)$	Poisson-only: $n = 1000$ photons	20.7362	25.9935
$\Gamma_2(x)$	Poisson-only: $n = 500$ photons	20.7138	25.9045
$\Gamma_2(x)$	Poisson-only: $n = 100$ photons	19.3611	25.3603
$\Gamma_2(x)$	Poisson-only: $n = 50$ photons	16.4748	24.2966
$\Gamma_2(x)$	Gaussian+Poisson: $\sigma = 0.001$, $n = 1000$	18.6196	21.3143
$\Gamma_2(x)$	Gaussian+Poisson: $\sigma = 0.01$, $n = 500$	18.3777	20.8228
$\Gamma_2(x)$	Gaussian+Poisson: $\sigma = 0.001$, $n = 50$	14.2750	19.6120
$\Gamma_2(x)$	Gaussian+Poisson: $\sigma = 0.1$, $n = 100$	15.2952	16.4092

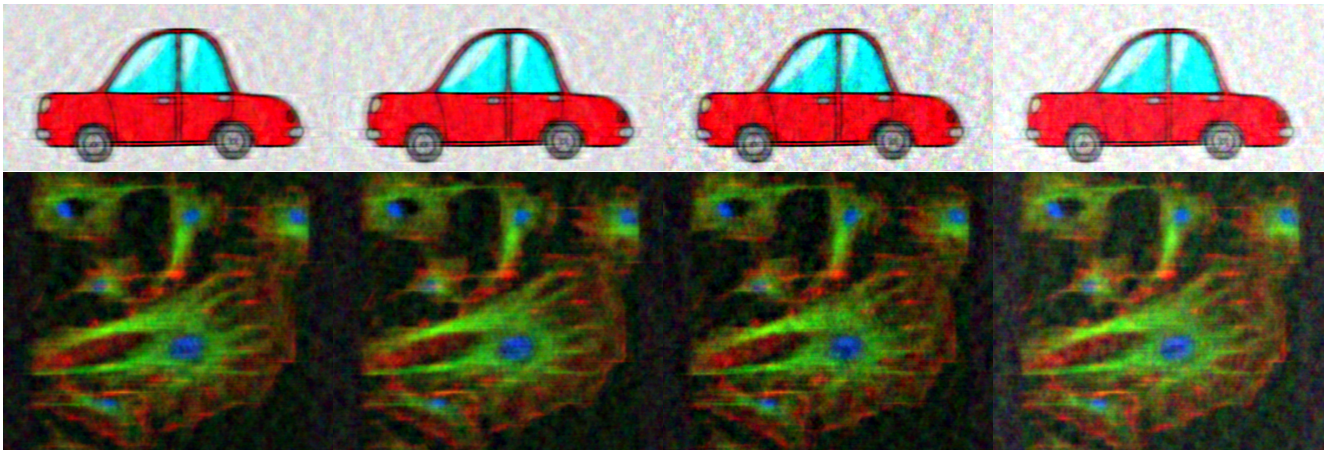
Table 1: Results of ADMM using regularization terms $\Gamma_1(x)$ and $\Gamma_2(x)$



(a) Images reconstructed under Gaussian-only Noise. Noise std. dev. σ from left to right: 0.001, 0.01, 0.1, 0.15

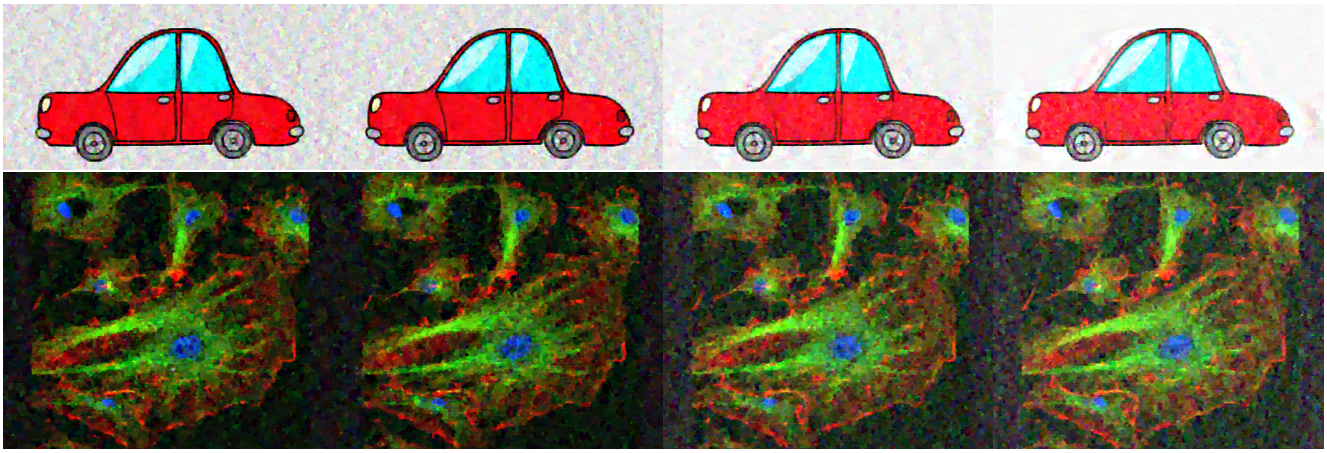


(b) Images reconstructed under Poisson-only Noise. photon count n from left to right: 1000, 500, 100, 50

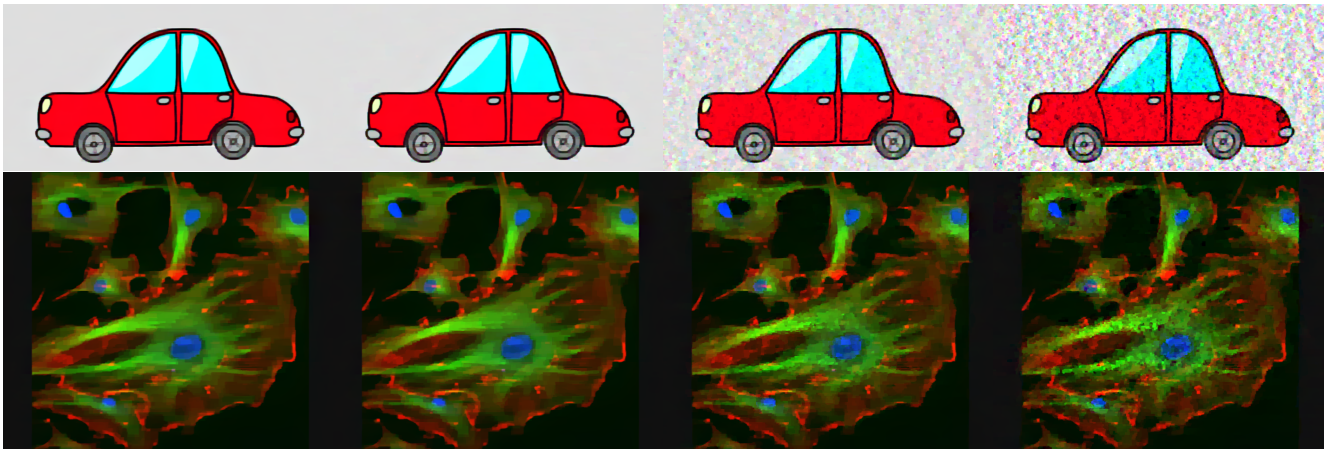


(c) Images reconstructed under mixed Gaussian-Poisson Noise. Parameters $[\sigma, n]$from left to right: $[.001, 1000]$, $[.01, 500]$, $[.001, 50]$, $[.1, 100]$

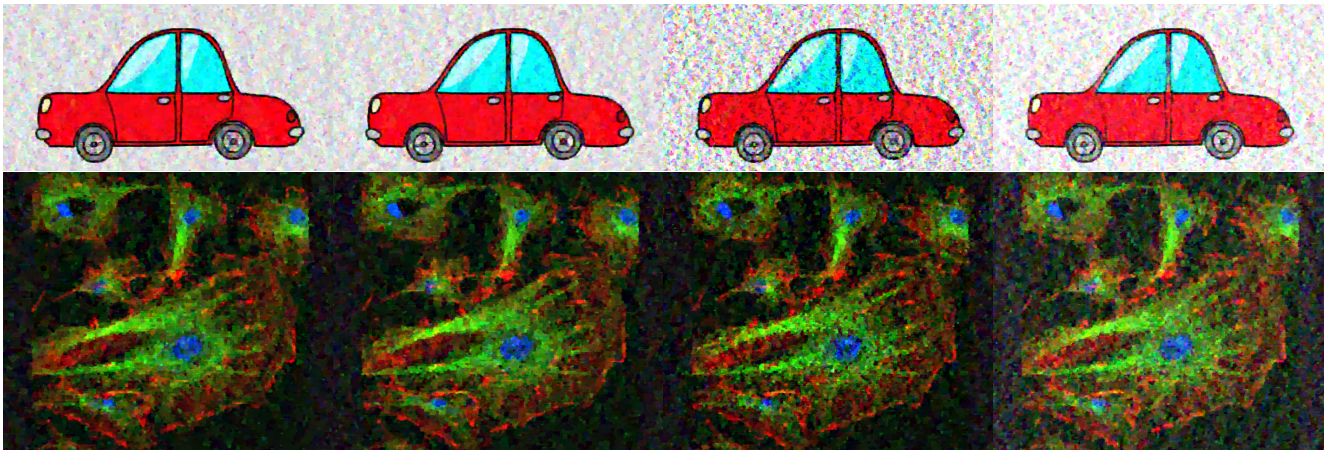
Figure 3: Results of ADMM using the Γ_1 regularization function with $\rho = 10$, $\lambda = .001$



(a) Images reconstructed under Gaussian-only Noise. Noise std. dev. σ from left to right: 0.001, 0.01, 0.1, 0.15



(b) Images reconstructed under Poisson-only Noise. photon count n from left to right: 1000, 500, 100, 50



(c) Images reconstructed under mixed Gaussian-Poisson Noise. Parameters $[\sigma, n]$from left to right: $[.001, 1000]$, $[.01, 500]$, $[.001, 50]$, $[.1, 100]$

Figure 4: Results of ADMM using the Γ_2 regularization function with $\rho = 10$, $\lambda = .15$