Image Deblurring and Denoising with Various Fidelity Terms and Regularizers

Kelsey Maass, Samuel Rudy, Kevin Mueller, and Riley Molloy
University of Washington

Abstract

1   Introduction

Image denoising and deblurring problems can be modeled with the linear system

$$Ax + w = b, \tag{1}$$

where b is the observed image, w is some unknown noise, and x is the true image we would like to recover. For a blurred image we let A represent the blurring process, and for noisy images we let A be the identity matrix. For many inverse problem applications the least squares approximation,

$$\hat{x}_{LS} = \arg\min_x \|Ax - b\|_F^2, \tag{2}$$

gives a reasonable solution. Unfortunately, this approach doesn't provide any new information for the denoising problem, and for image deblurring problems A is often ill-conditioned, so the solution is contaminated by round-off error and amplified noise. To illustrate this, we consider the pepper image below. If we blur the image and compute the naive solution $x = A^{-1}b$, the result is obscured by round-off error in the form of ringing artifacts. The result is even worse when we add noise.
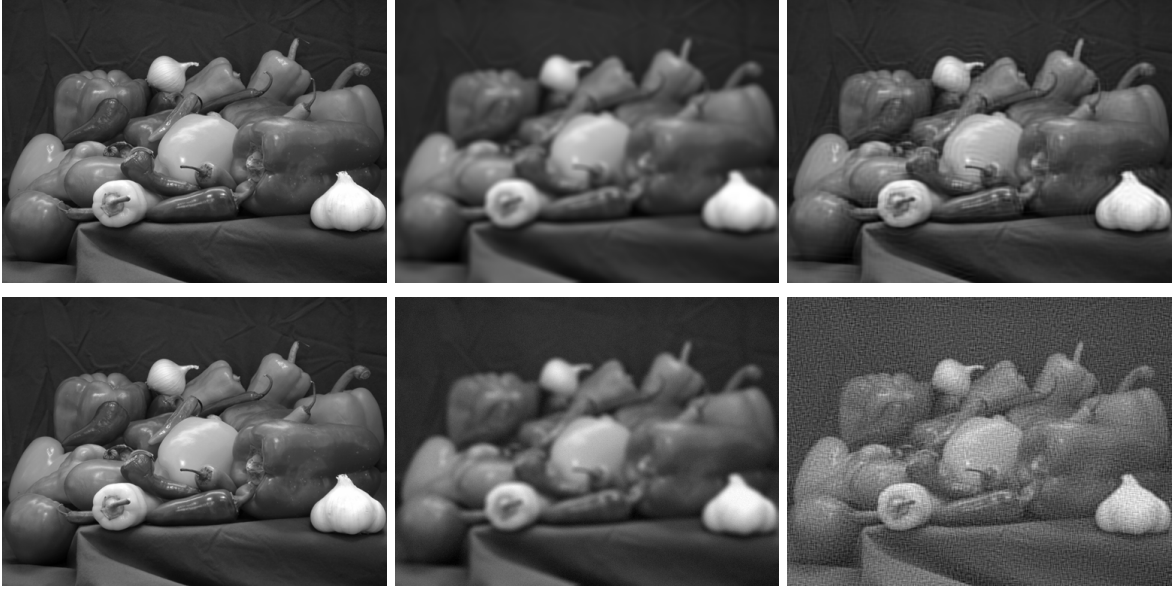
Figure 1: In the top row we observe the naive solution applied to a blurred image, including the true image x (left), the blurred image b (middle), and the recovered image $A^{-1}b$. On the bottom row, we see the effects of adding noise illustrated by the true image x (left), the blurred and noisy image $b-w$ (middle), and the recovered image $A^{-1}(b-w)$.

In order to stabilize the solution, a variety of regularizers can be used which exploit features of the true image such as smoothness or sparsity in a wavelet domain. This results in the problem formulation

$$\hat{x} = \arg\min_x \|Ax - b\|_F^2 + \lambda R(x), \tag{3}$$

where the parameter $\lambda > 0$ is chosen to balance the tradeoff between fidelity to the model and the assumed feature. In this paper we compare two choices for the regularization term,

Total variation regularization: $\qquad \hat{x}_{TV} = \arg\min_x \|Ax - b\|_F^2 + 2\lambda \text{TV}(x),$ $\qquad$ (4)

and ℓ_1 regularization: $\qquad \hat{x}_{\ell_1} = \arg\min_x \|Ax - b\|_F^2 + \lambda\|Wx\|_1,$ $\qquad$ (5)

which are explored by Beck and Teboulle in [1] and [2] respectively.

It is also known that the quadratic penalty is extremely sensitive to outliers. While this does not pose problems for images with Gaussian noise, which may result from atmospheric turbulence, it becomes an issues for images with noise from heavier tailed distributions, such as the Student's t-distribution, which could result from a dirty camera lens. Therefore we also consider different choices for the fidelity term which are more robust to outliers than the quadratic penalty, including the Huber norm

$$h_\gamma(x) = \min_y \frac{1}{2}\|x - y\|_F^2 + \gamma\|y\|_1, \tag{6}$$

and the function

$$g_\gamma(x) = \frac{1}{\gamma}\sum_i \log\left(\cosh\left(\gamma x_i\right)\right). \tag{7}$$

2

Penalty Functions for Fidelity Term

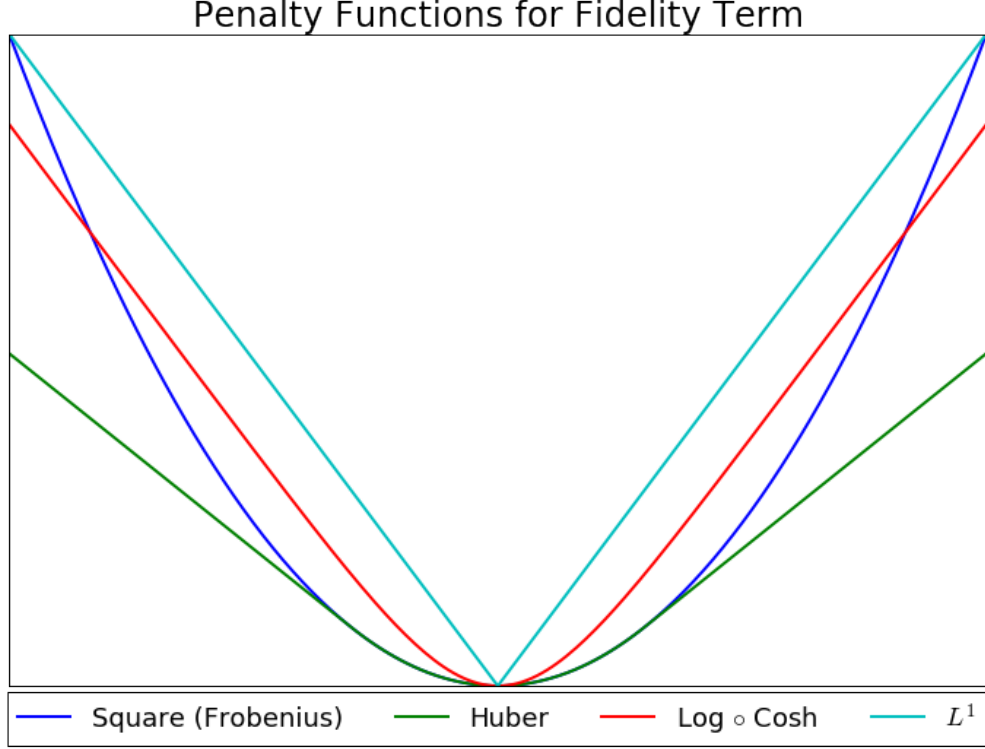| Square (Frobenius) | Huber | Log ∘ Cosh | L^1 |

Figure 2: al;sdfj?

2 Problem Formulation

The general approach to the image deblurring and denoising problem can now be stated as a minimization problem of the form [3]

$$\min_x f(Ax - b) + \lambda R(x), \tag{8}$$

where Ax is a discrete convolution of the true image with a blur kernel, the *fidelity term* $f(Ax - b)$ measures how well the recovered image complies with the linear model, and $R(x)$ is our chosen *regularization*. The parameter $\lambda > 0$ is chosen to balance the tradeoff between fidelity to the model and noise sensitivity. Furthermore, if we assume that the pixels of our image satisfy $x_{i,j} \in [0, 1]$, we can add an additional term to the objective function

$$L_b(x) = f(Ax - b) + \lambda R(x) + \delta(x|[0, 1]), \tag{9}$$

where $\delta(x|[0, 1])$ is the indicator function for the set $[0, 1]$.

2.1 Blur Operators

The blurring matrix A is built from two ingredients: the blur kernel, which specifies how each pixel spreads out in the blurred image, and the boundary conditions, which specify our assumptions about the scene just outside of our observed image. For our examples, we use a 9×9 Gaussian blur kernel with a standard deviation of one. The Gaussian kernel is popular in blurring applications, and can be used to simulate atmospheric turbulence.
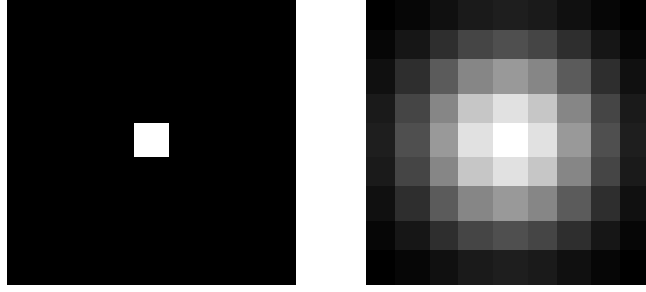
Figure 3: A single pixel before (left) and after blurring (right). The image on the right is called the *point spread function*, or the blur kernel. In this example the blur kernel computed by generating a 9×9 Gaussian with mean zero and standard deviation one, normalized so that the pixels sum to one.

In order to blur or deblur pixels near the border of our image, we need to decide how to represent the pixels just outside of our image boundaries, which we do not have. If we assume that these pixels mirror those inside, then we impose reflexive boundary conditions. For example, letting X be the pixels within our given image, we can construct an extended image X_{ext} with

$$X_{lr} = \texttt{fliplr}(X) \quad X_{ud} = \texttt{flipud}(X) \quad X_x = \texttt{fliplr}(X_{ud})$$

$$X_{ext} = \begin{bmatrix} X_x & X_{ud} & X_x \\ X_{lr} & X & X_{lr} \\ X_x & X_{ud} & X_x \end{bmatrix}$$

For spatially invariant and doubly symmetric blur kernels, these boundary conditions result in a blurring matrix that can be diagonalized by the orthogonal two-dimensional discrete cosine transform

$$A = C^T \Lambda C, \tag{10}$$

where the eigenvalues of A are determined by the blur kernel. There are two advantages of representing the blur operator in this way. First, we do not have to construct the matrix explicitly. Instead, we can take advantage of the efficient built-in discrete cosine transform function to compute our blurred image:

$$Ax = C^T \Lambda C x = \texttt{idct2}(\Lambda \texttt{dct2}(x)) = b. \tag{11}$$

Second, when our fidelity function is given by $f(Ax - b) = \|Ax - b\|_2^2$, we can exploit the fact that C is orthogonal to evaluate the term more efficiently.

$$\begin{aligned} f(Ax - b) &= \|Ax - b\|_2^2 \\ &= \|C^T \Lambda C x - b\|_2^2 \\ &= \|\Lambda C x - C b\|_2^2 \\ &= \|\Lambda \hat{x} - \hat{b}\|_2^2 \end{aligned}$$

2.2   Wavelet Regularization

The wavelet regularization deblurring model, as seen in [2], can be formulated in the form of (8) as

$$\min_x \|Ax - b\|_F^2 + \lambda \|Wx\|_1, \tag{12}$$

where $\|\cdot\|_F$ is the Frobenius norm, $\lambda > 0$ is a regularization parameter, and W corresponds to a wavelet trasform of a given wavelet type. The primary motivation for this approach is that most images have a sparse representation in the wavelet domain that can be easily exploited by ℓ_1 regularization.

Our approach considers a more general problem

$$\min_x f(Ax - b) + \lambda \|Wx\|_1, \tag{13}$$

where $f : \mathbb{R}^{m \times n} \to [0, \infty)$ is some Lipschitz differentiable functional which gives a measurement of the size of the fidelity term. We specifically work with the cases

$$f(x) = \begin{cases} \|x\|_F^2 \\ h_\gamma(x) \\ g_\gamma(x), \end{cases} \tag{14}$$

4

where h_γ and g_γ are as in (6) and (7) respectively.

2.3 Total Variation Regularization

The usual Total-Variation deblurring model, as seen in [1] can be formulated in the form of (8) as

$$\min_x \|Ax - b\|_F^2 + 2\lambda \text{TV}(x) + \delta(x|[0,1]), \tag{15}$$

where $\|\cdot\|$ is taken as either a Frobenius norm or a 2-norm, depending on applications, $\lambda > 0$ is a regularization parameter, and $\text{TV}(x)$ is the Total-Variation semi-norm. We include the indicator function to ensure the resulting image is within the range of pixel values we are considering. Two similar choices exist for the TV-norm: the so-called isotropic type, and the l_1 type. In this work, we work exclusively with the l_1-based TV-norm, defined as

$$TV_{l_1}(x) = \sum_{i=1}^{m-1}\sum_{j=1}^{n-1} (|x_{i,j} - x_{i+1,j}| + |x_{i,j} - x_{i,j+1}|) + \sum_{i=1}^{m-1} |x_{i,n} - x_{i+1,n}| + \sum_{j=1}^{n-1} |x_{m,j} - x_{m,j+1}|,$$

for $x \in \mathbb{R}^{m \times n}$, and where the reflexive boundary conditions

$$x_{m+1,j} - x_{m,j} = 0, \text{ for all } j$$
$$x_{i,n+1} - x_{i,n} = 0, \text{ for all } i$$

are assumed. However, it is a very simple adjustment to use the isotropic definition of total variation and all of the same methods still apply. Our approach considers a more general problem

$$\min_x f(Ax - b) + \lambda \text{TV}_{l_1}(x) + \delta(x|[0,1]), \tag{16}$$

where $f : \mathbb{R}^{m \times n} \to [0, \infty)$ is any functional with Lipschitz continuous gradient which gives a measurement of the size of the fidelity term. The specific fidelity terms which we worked with were as listed in (14).

3 Algorithms

A simple way to solve for the real image x give $g(x) = 0$ is to apply the gradient descent algorithm,

$$x_k = x_{k-1} - t_k \nabla f(x_{k-1}), \tag{17}$$

for suitable stepsize $t_k > 0$. When g is nontrivial and nonsmooth (the usual case), we can rewrite this as a proximal mapping,

$$\text{prox}_g(x_{k-1} - \nabla f(x_{k-1})) = \min_x \left(\frac{1}{2}\|x - x_{k-1} - \nabla f(x_{k-1})\|_2^2 + \lambda g(x) \right) \tag{18}$$

The advantage of this form is that many nonsmooth regularizers have a close formed solution for their prox. This approach is referred to as a "proximal gradient" method. We use this setup for both cases of regularizers, and cover the details of their implementation in the following subsections.

3.1 Wavelet Regularization

3.2 Total Variation Regularization

Optimization of the objective function (16) is based on the proximal gradient algorithm and expoits the idea of momentum used in FISTA as well as an extra function evaluation at each iteration in order to ensure a non-increasing objective function. The proximal gradient step is given as follows,

$$x^{k+1} = \text{prox}_{\alpha^{-1}(\lambda \text{TV}(\cdot) + \delta_{[0,1]})} (\underbrace{x^k - \alpha^{-1} A^T \nabla f(Ax^k - b)}_{u^k})$$
$$= \arg\min_z \left(\|u^k - z\|_F^2 + \alpha^{-1}\lambda TV(z) + \delta(z|[0,1]) \right)$$
$$= P_{[0,1]} \left(\arg\min_z \left(\|u^k - z\|_F^2 + \alpha^{-1}\lambda TV(z) \right) \right)$$

Here the learning rate α is taken to be no larger than than the multiplicative inverse of the Lipschitz constant of ∇f. Note that in order to evaluate this expression we must solve a denoising problem with input $u^k = x^k - \alpha^{-1} A^T \nabla f(Ax^k - b)$. Of particular note is that regardless of the original fidelity term we have a Frobenius norm fidelity term on the new denoising problem. This allows for the easy manipulation of the original fidelity function, so long as it is convex and has Lipschitz gradient.

The resulting denoising problem with Frobenius norm fidelity function may be solved rapidly using the method developed in [1]. The *TV* function is shown to have a dual representation as a trace and the relationship between trace and Frobenius norm is heavily exploited to develop a dual formulation of the denoising problem. We present their method here, starting with a few new definitions which will be necessary.

$$\mathcal{P} = \{(p,q) \in \mathbb{R}^{(m-1)\times n} \times \mathbb{R}^{m\times(n-1)} : |p_{i,j}| \le 1, |p_{i,j}| \le 1\}$$

$$\mathcal{L} : \mathbb{R}^{(m-1)\times n} \times \mathbb{R}^{m\times(n-1)} \to \mathbb{R}^{m\times n} \text{ such that } \mathcal{L}(p,q)_{i,j} = p_{i,j} + q_{i,j} - p_{i-1,j} - q_{i,j-1}$$

$$\text{for } i = 1, \ldots, m, \ j = 1, \ldots, n \text{ and } p_{0,j} = p_{m,j} = q_{i,0} = q_{i,n} = 0$$

Using these new definitions we claim without proof that the total variation functional may be written as,

$$TV(x) = \max_{(p,q)\in\mathcal{P}} \text{Tr}\big(\mathcal{L}(p,q)^T x\big) \tag{19}$$

we may now exploit the relationship between trace and Frobenius norm to obtain the dual problem for Frobenius denoising using total variation regularizartion.

$$
\begin{aligned}
\min_{x\in[0,1]} \|x - b\|_F^2 + 2\lambda\text{TV}(x) &= \min_{x\in[0,1]} \max_{(p,q)\in\mathcal{P}} \|x - b\|_F^2 + 2\lambda\,\text{Tr}\big(\mathcal{L}(p,q)^T x\big) \\
&= \max_{(p,q)\in\mathcal{P}} \min_{x\in[0,1]} \|x - b\|_F^2 + 2\lambda\,\text{Tr}\big(\mathcal{L}(p,q)^T x\big) \\
&= \max_{(p,q)\in\mathcal{P}} \min_{x\in[0,1]} \|x\|_F^2 + \|b\|_F^2 - 2\,\text{Tr}\big(x^T b\big) + 2\lambda\,\text{Tr}\big(\mathcal{L}(p,q)^T x\big) \\
&= \max_{(p,q)\in\mathcal{P}} \min_{x\in[0,1]} \|x\|_F^2 + \|b\|_F^2 - 2\,\text{Tr}\big(x^T(b - \lambda\mathcal{L}(p,q))\big) \\
&= \max_{(p,q)\in\mathcal{P}} \min_{x\in[0,1]} \underbrace{\|x\|_F^2 + \|b - \lambda\mathcal{L}(p,q)\|_F^2 - 2\,\text{Tr}\big(x^T(b - \lambda\mathcal{L}(p,q))\big)}_{=\|x-(b-\lambda\mathcal{L}(p,q))\|_F^2} - \|b - \lambda\mathcal{L}(p,q)\|_F^2 + \|b\|_F^2 \\
&= \max_{(p,q)\in\mathcal{P}} \min_{x\in[0,1]} \|x - (b - \lambda\mathcal{L}(p,q))\|_F^2 - \|b - \lambda\mathcal{L}(p,q)\|_F^2 + \|b\|_F^2
\end{aligned}
$$

Here the switching of min and max is permitted due to the convexity of the objective in x and concavity in p and q [1]. Note that the problem of maximizing over x is simply the projection of each index of $b - \lambda\mathcal{L}(p,q)$ onto the set $[0,1]$. This gives the optimality condition for x in terms of p and q which we plug back in to obtain the dual problem.

$$(p^*, q^*) = \arg\max_{(p,q)\in\mathcal{P}} \big\|P_{[0,1]}(b - \lambda\mathcal{L}(p,q)) - (b - \lambda\mathcal{L}(p,q))\big\|_F^2 - \|b - \lambda\mathcal{L}(p,q)\|_F^2 + \|b\|_F^2$$

$$x = P_{[0,1]}(b - \lambda\mathcal{L}(p^*, q^*))$$

Note that in the dual problem the objective is Lipschitz differentiable. We claim without proof that the gradient has Lipschitz constant $L \le 16\lambda^2$ and refer the reader to [1] for proof. This smoothness allows us to implement a projected gradient algorithm.

We implement the above optimization scheme by using a slightly optimized method referred to as Monotone FISTA or MFISTA. In FISTA, the objective function values are not guaranteed to be nonincreasing, so MFISTA requires the evaluation of the objective function to check for monotonicity. *talk about how FGP is fista version of the projection equation that's not here yet*

$$\textbf{MFISTA}(b, f, \lambda)$$
$$\quad y^1 = x^0 = b; \, t^1 = 1$$
$$\quad \alpha \geq Lip(\nabla f)$$
$$\quad \textbf{for } k = 1 : N \textbf{ do}$$
$$\quad\quad u^k = y^k - \frac{A^T \nabla f(Ay^k - b)}{\alpha}$$
$$\quad\quad z^k = FGP(u^k, \frac{\lambda}{2\alpha})$$
$$\quad\quad x^k = \underset{x \in \{x^{k-1}, z^k\}}{\operatorname{argmin}} L_b(x)$$
$$\quad\quad t^{k+1} = \frac{1 + \sqrt{1 + 4t^{k^2}}}{2}$$
$$\quad\quad y^{k+1} = x^k + \frac{t^k}{t^{k+1}}(z^k - x^k)$$
$$\quad\quad\quad + \frac{t^{k-1}}{t^{k+1}}(z^k - x^k)$$
$$\quad \textbf{end for}$$
$$\quad \textbf{return } x^N$$

$$\textbf{FGP}(b, \lambda)$$
$$\quad (r^1_{ij}, s^1_{ij}) = (p^0_{ij}, q^0_{ij}) = 0; \, t^1 = 1$$
$$\quad \textbf{for } k = 1 : N \textbf{ do}$$
$$\quad\quad (p^k, q^k) = P_{\mathcal{P}}\left((r^k, s^k) - \frac{\mathcal{L}^T P_{[0,1]}(b - \lambda\mathcal{L}(r^k, s^k))}{8\lambda}\right)$$
$$\quad\quad t^{k+1} = \frac{1 + \sqrt{1 + 4t^{k^2}}}{2}$$
$$\quad\quad (r^k, s^k) = (p^k, q^k) + \frac{t^k - 1}{t^{k+1}}(p^k - p^{k-1}, q^k - q^{k-1})$$
$$\quad \textbf{end for}$$
$$\quad \textbf{return } P_{[0,1]}(b - \lambda\mathcal{L}(p^N, q^N))$$

3.3 Wavelet Regularization

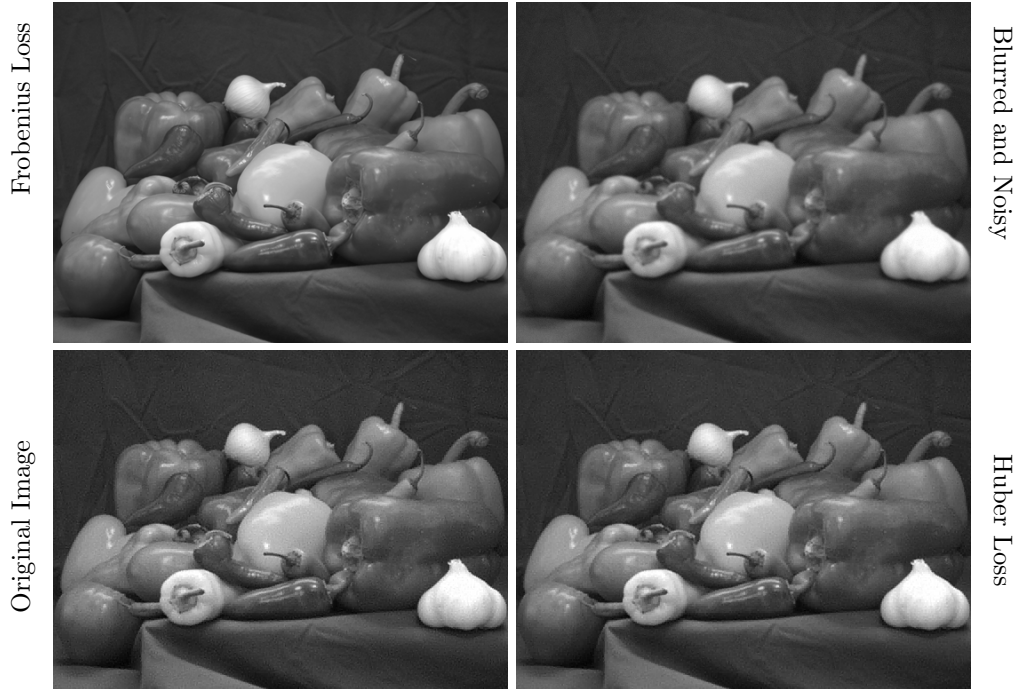# 4 Examples

4.1 Wavelet Regularization

Figure 4: ℓ_1 Wavelet algorithm performed on an image with Gaussian noise (magnitude 1×10^{-3}) for the two different fidelity terms of Frobenius and Huber loss. This method used the Haar wavelet for deblurring.
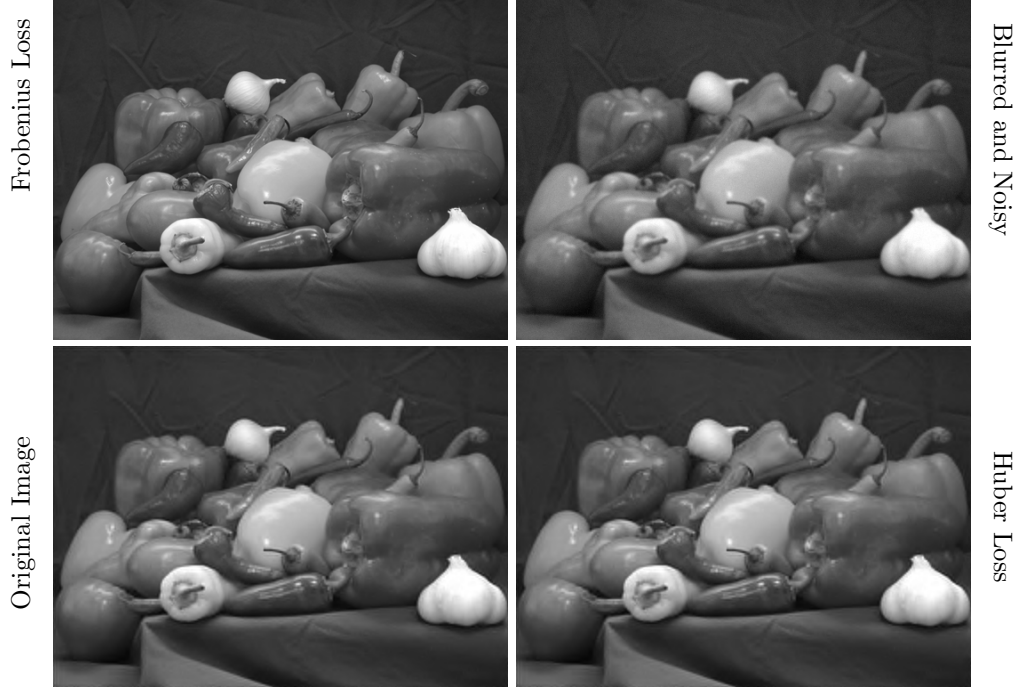
Figure 5: ℓ_1 Wavelet algorithm performed on an image with Gaussian noise (magnitude 1×10^{-3}) for the two different fidelity terms of Frobenius and Huber loss. This method used the Daubechies wavelet for deblurring.
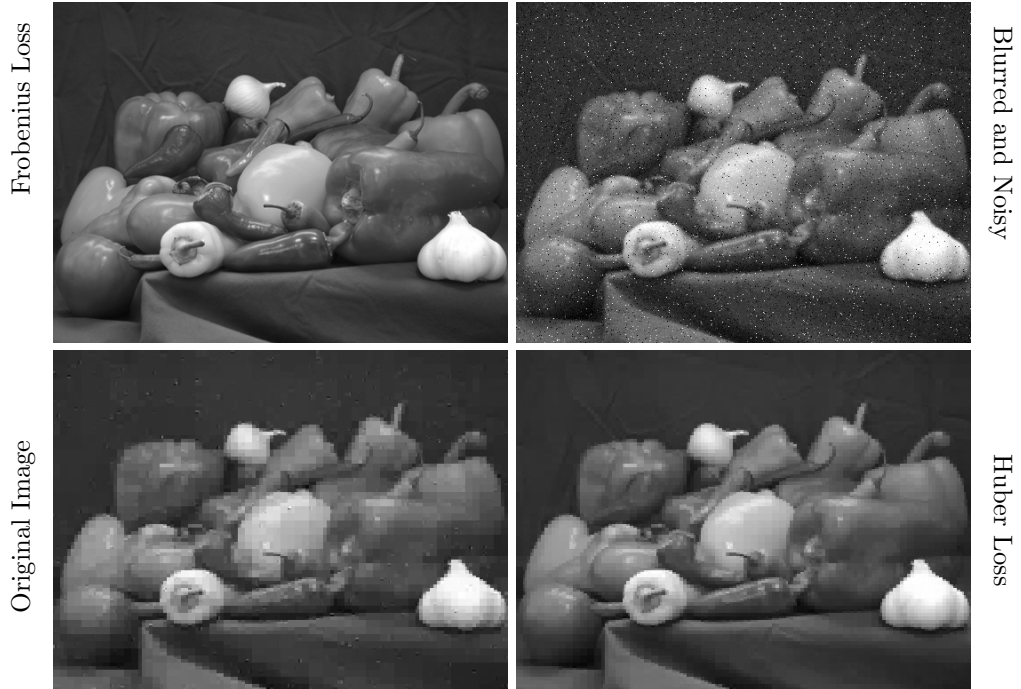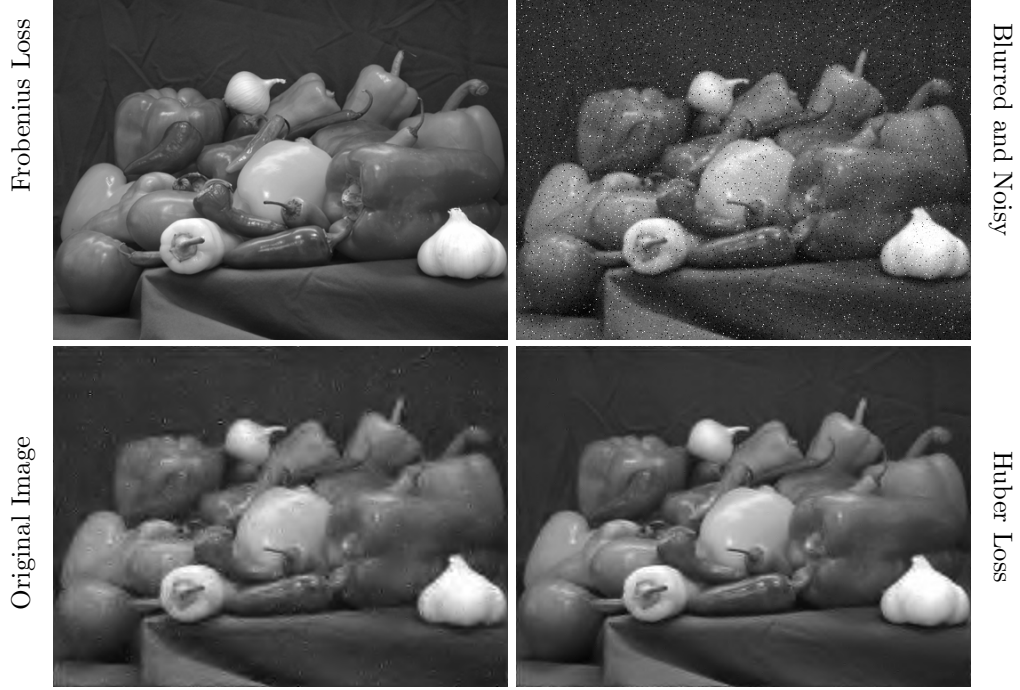

Figure 6: ℓ_1 Wavelet algorithm performed on an image with Student's t noise (magnitude 1×10^{-4}) for the two different fidelity terms of Frobenius and Huber loss. This method used the Haar wavelet for deblurring.

Figure 7: ℓ_1 Wavelet algorithm performed on an image with Student's t noise (magnitude 1×10^{-4}) for the two different fidelity terms of Frobenius and Huber loss. This method used the Daubechies wavelet for deblurring.

4.2 Total Variation Regularization

We demonstrate some results of denoising and deblurring with the Total Variation regularization on real images. As previously shown, each image was given a small amount of either Gaussian or Student's t noise and blurred. Two different fidelity functions were used: the squared Frobenius norm, and the Huber norm. Figure 8 demonstrates the results using Gaussian noise, and Figure 9 demonstrates the results using Student's t noise with one degree of freedom. For the Gaussian noise, we set $\lambda = .001$, and $\gamma = .02$ (see TV-regularization in Problem Formulation). For Student's t noise, we set $\lambda = .002$, and $\gamma = .02$.
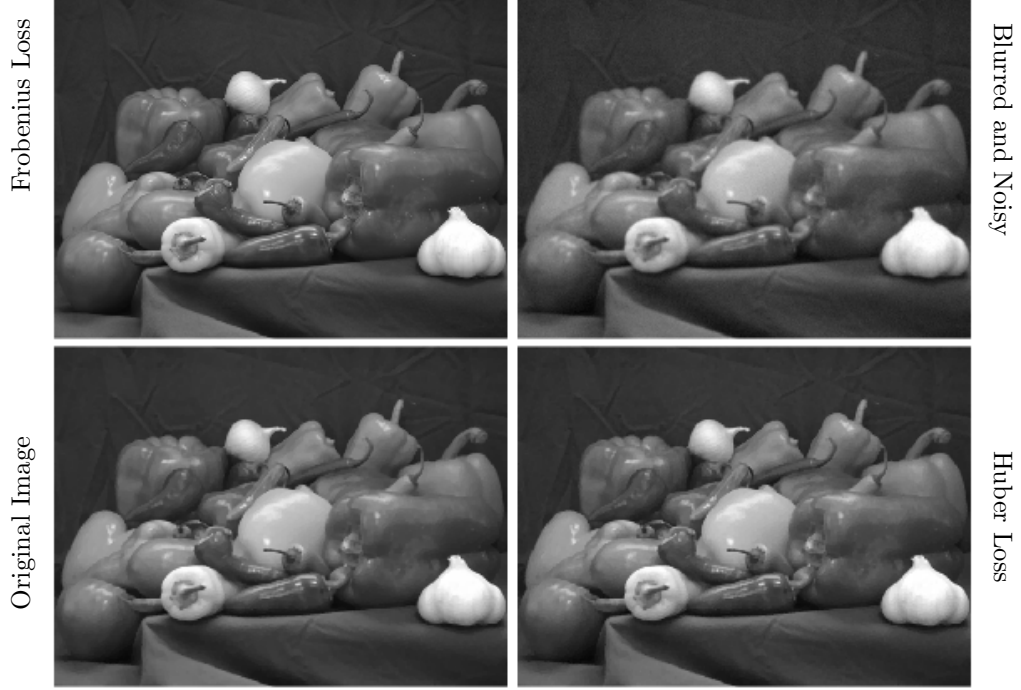
9

Figure 8: Total Variation algorithm performed on an image with Gaussian noise (magnitude 1×10^{-3}) for the two different fidelity terms of Frobenius and Huber loss.
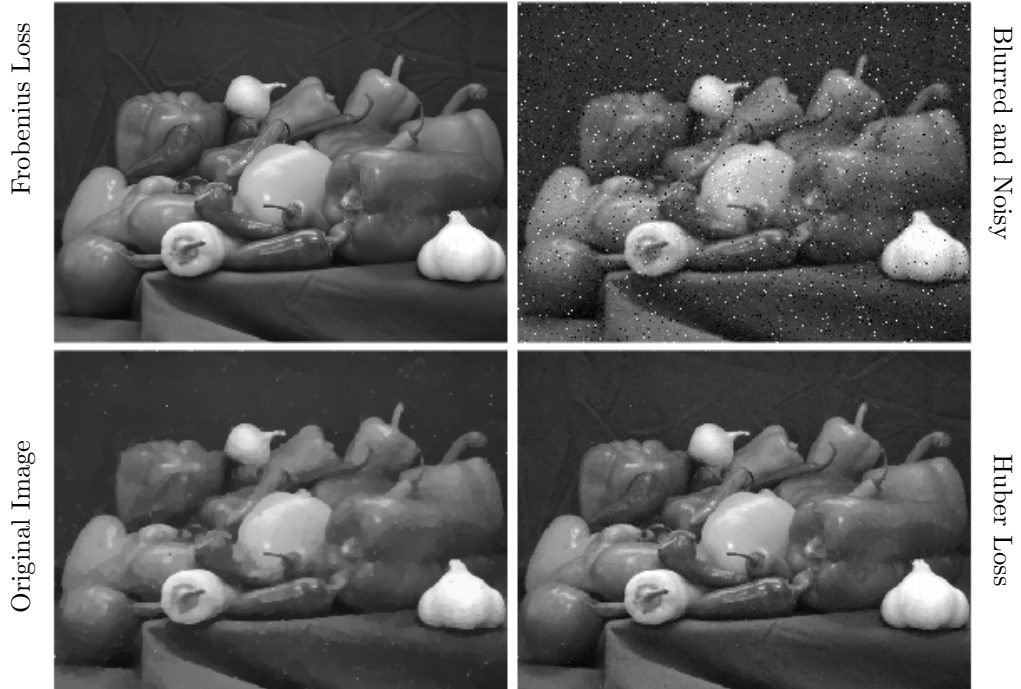

Figure 9: Total Variation algorithm performed on an image with Student's t noise (magnitude 1×10^{-4}) for the two different fidelity terms of Frobenius and Huber loss.

4.3 The $\log \circ \cosh$ Penalty Function

Recall that for large values of γ the function $\gamma^{-1} \log(\cosh(\gamma \cdot))$ uniformly converges to the absolute value function. Since it is also Lipschitz differentiable we are able to use it in our algorithm for TV regularized denoising and deblurring with high values of γ to approximate how the standard L^1 norm would perform without needing to worry about how to optimize it.
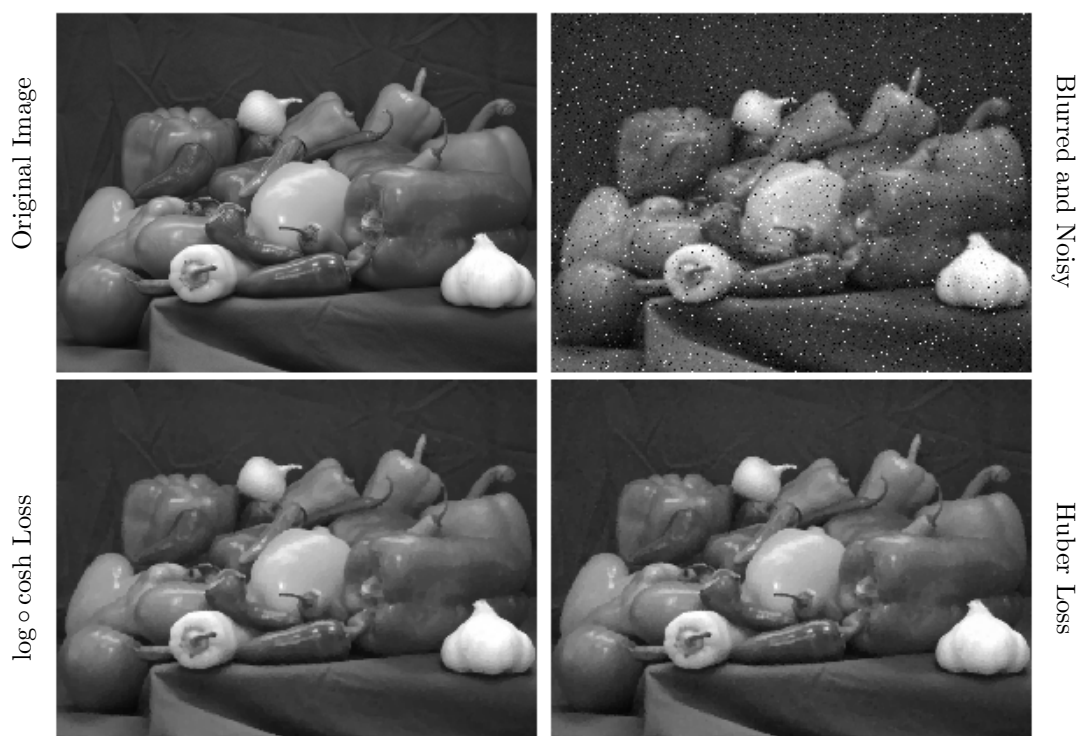

Figure 10: A comparrison between the $\log \circ \cosh$ and Huber Fidelity Terms

To do: -it seems to perform better -it also takes longer since the step size decreases like 1/gamma
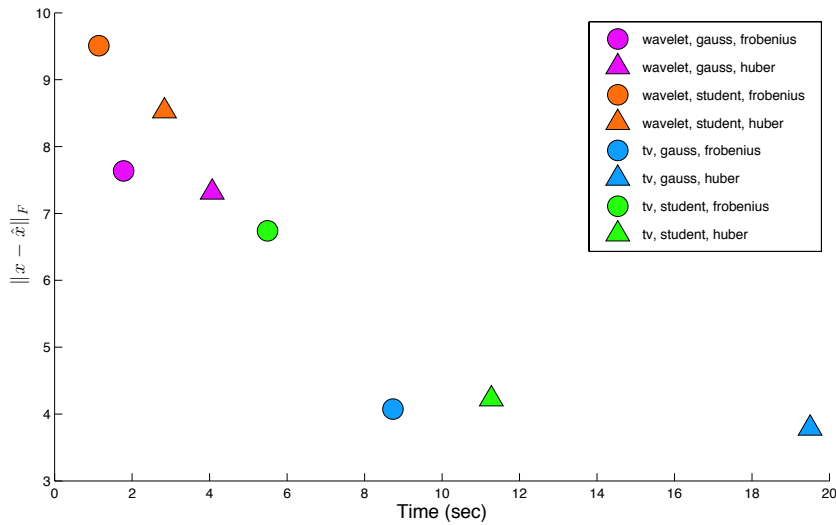
5 Discussion and Conclusions

Figure 11: Timing information and the Frobenius norm error between the true and observed image until the relative change in the objective function is less than .001.

Figure 11 shows a rough comparison of our methods by displaying the time complexity and Frobenius error for different fidelity functions, regularization techniques and noise-types. It is clear that TV regularization outperformed wavelet regularization at the price of additional computational complexity. This is not surprising due to TV's extra requirement of solving an inner optimization problem for denoising the image. It can also be seen that student noise generally required less time than Gaussian noise but at the expense of higher error. Finally, applying the Huber for student noise improved results drastically for a slight increase in computation time.

In conclusion, we present a survey of regularization and fidelity functions for the image denoising/deblurring problem. In particular, we compare regularization of the 1-norm in the wavelet domain to TV regularization. Furthermore we explore the affect of using the Huber as a fidelity function and potential improvements it can have when applied to heavy tailed noise. Our results show that TV regularization performs better for high amounts of noise, but the wavelet domain is well suited for smaller amounts of noise due to computational speed improvements.

In our research we came across many challenges and avenues for future work. It is clear from **??**, that the correct choice of the regularization parameter is very important for maximizing results. However, the correct value varies for different classes of images. Therefore, there is a high need for finding the correct value for the regularization parameter across images. Additionally, finding an appropriate error metric can be a difficult challenge when quantitatively analyzing performance for the methods shown. The Frobenius norm of the error is the most common metric, but does can do a poor job of correctly evaluating the improvements from denoising and deblurring algorithms due to large structural changes of the images.

References

[1] A. Beck and M. Teboulle. Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. *IEEE Transactions on Image Processing*, 18(11):2419–2434, November 2009.

[2] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal of Imaging Sciences*, 2(1):183–202, 2009.

[3] J.G.Nagy P.C. Hansen and D.P. O'Leary. *Deblurring Images: Matrices, Spectra, and Filtering.* SIAM, Philadelphia, 2006.