A Comparison and Compilation of Spatial Image Priors for Image Deconvolution

Amit Kohli
Stanford University
450 Serra Mall, Stanford, CA 94305
apsk14@stanford.edu

Abstract

Image priors are critical tools used to turn ill-posed deconvolution problems into forms that have more tractable solutions. However, as these priors become more and more ubiquitous, it is important to thoroughly test, classify, and organize these priors for various classes of images under different conditions. This treatment of image priors is fundamental for robust and novel image estimation to become an easily usable and intuitive technology that anyone and everyone can use. In this paper I simply demonstrate a small example of how I feel priors can be compared, generalized, and compiled together. Specifically, I explore three spatial priors, Total Variation, Laplacian, and Frobenius norm of Hessian, applied to natural images using the ADMM solution model for non-blind image deconvolution. I find that these norms operate well as a basis for optimizing natural images. Additionally, I combine them into a single function for the sake of showing how priors may be combined into a general technology.

1. Introduction

Often, new image priors are created for and demonstrated only for specific applications and for a specific audience. As a result, communities outside of the research community have a limited knowledge of priors. In effect, there exists a knowledge gap that results less accessibility of the priors for commercial and interdisciplinary use. Bridging this gap will increase the range of impact of research that develops priors and allow for a larger set of individuals to use and understand priors in potentially new and creative ways.

To give an example of how one might increase the accessibility of knowledge on image priors, I perform an analysis on and combine three spatial image priors into a single, simple function. I explain the tradeoffs associated with using one prior over another and give a description of how my general deblur function can be used effectively.

1.1. Image Priors Background

In the spirit of making information more accessible, I will give a brief background on the notion of an image prior. Image priors are essentially assumptions that one makes about the characteristics of an image. These assumptions aid the process of estimating a good, clear image from an image that is corrupted. Priors make assumptions about a wide variety of image characteristics and are most dependent on what type of image is being processed.

These priors are usually incorporated into existing methods for solving image estimation problems. Generally, these methods simply involve the attempt to invert the process in which the image was corrupted, generally referred to as the 'image formation model'. Most often, this problem of formation inversion is ill-posed and difficult to solve, hence the need for a prior. The prior is formatted as a mathematical expression which can be jointly optimized with the desire to invert the image corruption. There is usually a tuning parameter that allows one to tradeoff between inverting the corruption and meeting the assumption of the prior.

1.2. Spatial Image Priors

The class of priors I explore in this paper are spatial image priors. These priors place an assumption on how the intensities are structured relative to the location on the image. For example, each of the priors I consider assume that the image is smooth, meaning that there are small changes in the intensities between pixels that are close to each other. The priors I consider are the **Total Variation (TV)** prior, a first order operator that assumes piecewise smoothness, the **Laplacian** prior, a second order prior that assumes smoothness, and the **Frobenius norm of Hessian (FOH)** prior, a second order prior that assumes smoothness.

To motivate why this might work for natural images I include each of the three priors applied directly to the reference image; see Figure 1. Clearly each of the smoothness measures has its intensity sparsely distributed. Additionally, we see that the actual measurement of the prior objective is much smaller for the FOH prior and allows us to hypothesize that this prior will perform better for the example image. However, we also see that the FOH prior includes far more detail and may be more computationally more expensive.

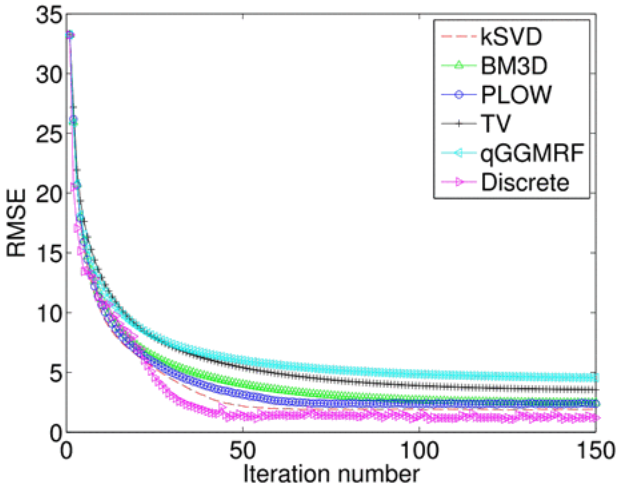
	Reference	TV	Laplacian	FOH
Objective Values:		3,180	4,488	65.2

Figure 1: Magnitudes of Spatial Priors with values of each prior objective

2. Related Work

An enormous amount of work has been done in the development of image priors. However, there is less literature in the comparison, classification, and compilation of priors. One inspiration for this paper was the work done by Venkatakrishnan et al. [1]. In their paper, they perform a thorough treatment of six different priors which they analyze, generalize, and combine into a framework. They then express this framework as a simple tool that can potentially have easy software integration. Fundamentally, their work helps make the priors they chose more accessible and useable. I continue their work and apply a similar methodology in the context of applying spatial priors to natural images.

I follow Venkatakrishnan et al.'s methodology of comparing priors. In Figure 2 are two formats from their paper that are effective ways of displaying these comparisons. I will take a similar approach of comparing priors for a single image with each prior tuned to the optimal value of the tuning parameter. However, in addition to PSNR, I also include SSIM as a performance metric. Like Venkatakrishnan et al., I also look at convergence of the priors relative to each other to make statements about runtime and computational performance.



Algorithm	RMSE	β
K-SVD [4]	2.13	4.32
BM3D [5]	2.46	1.39
PLOW [7]	2.35	1.50
TV [21]	3.55	0.47
q-GGMRF [22]	4.58	0.28
Discrete Recon [23]	1.20	1.00

Figure 2: Venkatakrishnan et al.'s analysis of priors

3. Methodology

The way I carry out this study of priors is ultimately by implementing each prior, compiling them into a single deblur function, measuring their performance statistics, and comparing their results visually. Specifically, I perform the tests on a dataset of 24 images provided by Kodak for open use. Additionally, I run the tests on an example image for direct visual comparison between the priors under the same conditions. For each test I restrict the image corruption to Gaussian low pass filter and additive Gaussian noise. For each prior, I used a method outlined by a reference paper. Although I will include a general description of my particular implementation, interested readers should refer to my reference papers for a deeper exploration of each prior.

The general image formation model for each of the priors is an image convolved with a known blur kernel and added to by Gaussian noise. Generally, this can be expressed as:

$$b = Cx + \eta \qquad (1)$$

Where b is the measured, corrupted image, C is the blur operation, x is the true image that we desire to recover, and η is the noise term. Alternatively, we may write the blur as a convolution operation with blur kernel c:

$$b = c * x + \eta \tag{2}$$

The general solution form for this image formation model becomes an optimization problem:

$$minimize_x \frac{1}{2} \| Cx - b \|_2^2 + \lambda\Gamma(x) \tag{3}$$

Where $\Gamma(x)$ is the mathematical expression for the prior and λ is the tuning parameter. I now simply define the $\Gamma(x)$ for each prior and insert it into the problem to solve.

3.1. Total Variation Prior

The TV prior I implement is based on Rudin et al.'s paper [2] and specifically, as outlined in Professor Gordon Wetzstein's EE 367 course notes [3]. Here the prior is defined as

$$\Gamma(x) = |Dx|_1, \; D = \begin{bmatrix} D_x \\ D_y \end{bmatrix} \tag{4}$$

Where D_x and D_y are finite difference operators in the x and y direction, respectively. In my implementation this carried out with convolution kernels with:

$$\mathbf{D}_x\mathbf{x} = \text{vec}\,(d_x * x), \quad d_x = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$
$$\mathbf{D}_y\mathbf{x} = \text{vec}\,(d_y * x), \quad d_y = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{5}$$

3.2. Laplacian Prior

The Laplacian prior implementation is simply an extension of the TV prior implementation, only using the definition of the Laplacian operator.

$$\Gamma(x) = |Lx|_1, \; Lx = D_{xx}x + D_{yy}x \tag{6}$$

Where L is the Laplacian operator and D_{xx} and D_{yy} are second order difference operators. Just like the TV prior, the Laplacian can also be expressed with a convolution kernel:

$$Lx = \text{vec}\,(d_l * x), d_l = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \tag{7}$$

3.3. Frobenius Norm of Hessian Prior

The FOH prior implementation comes from Liu et al.'s [4] work in which they demonstrate an application of the FOH prior to 2D natural images. In their paper they convert the FOH into a sum of Euclidean norms:

$$\Gamma(x) = |Hx|_F = \sum_{i=1}^{MN} \| V_i x \|_2 , \tag{8}$$

$$V_i = \begin{bmatrix} D_{xx_i} \\ D_{xy_i} \\ D_{xy_i} \end{bmatrix} \in \mathbb{R}^3$$

Where Hx is the image Hessian, $D_{xx_i}x$ is a scalar that refers to second order difference operator evaluated at the ith pixel of x. Again, I implement this using convolution where:

$$D_{xx}x = \text{vec}(d_x * d_x * x)$$
$$D_{xy}x = \text{vec}(d_x * d_y * x) \tag{9}$$
$$D_{yy}x = \text{vec}(d_y * d_y * x)$$

3.4. Solution Model

All of these priors, when input into the general optimization problem, can be solved with ADMM [5]. The general ADMM solution form contains three iterative updates based on the augmented Lagrangian of the objective function. I will not derive these expressions for this paper. An interested reader may refer to the original implementations cited in this paper and additionally Boyd et al.'s [5] paper for further detail. All matrix operations are carried out in the Fourier domain using the convolution kernels specified in the previous section.

The ADMM formulations for the TV and Laplacian priors are very similar and will be treated as the same here. Similarly, my Matlab implementation uses the same ADMM function for both priors. It is as follows:

$$\mathbf{x} \leftarrow \text{prox}_{f,\rho}(\mathbf{v}) = \underset{\{\mathbf{x}\}}{\arg\min}\, L_\rho(\mathbf{x},\mathbf{z},\mathbf{y}) = \underset{\{\mathbf{x}\}}{\arg\min}\, f(\mathbf{x}) + \frac{\rho}{2}\|\mathbf{D}\mathbf{x} - \mathbf{v}\|_2^2, \quad \mathbf{v} = \mathbf{z} - \mathbf{u}$$
$$\mathbf{z} \leftarrow \text{prox}_{g,\rho}(\mathbf{v}) = \underset{\{\mathbf{z}\}}{\arg\min}\, L_\rho(\mathbf{x},\mathbf{z},\mathbf{y}) = \underset{\{\mathbf{z}\}}{\arg\min}\, g(\mathbf{z}) + \frac{\rho}{2}\|\mathbf{v} - \mathbf{z}\|_2^2, \quad \mathbf{v} = \mathbf{D}\mathbf{x} + \mathbf{u}$$
$$\mathbf{u} \leftarrow \mathbf{u} + \mathbf{D}\mathbf{x} - \mathbf{z} \tag{10}$$

The x update is

$$\tilde{\mathbf{x}} = \left(\mathbf{C}^T\mathbf{C} + \rho\mathbf{D}^T\mathbf{D}\right)^{-1}\left(\mathbf{C}^T\mathbf{b} + \rho\mathbf{D}^T\mathbf{v}\right) \tag{11}$$

The z update is

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

$$\mathcal{S}_\kappa(v) = \begin{cases} v-\kappa & v > \kappa \\ 0 & |v| \le \kappa \\ v+\kappa & v < -\kappa \end{cases} = (v-\kappa)_+ - (-v-\kappa) \tag{12}$$

Finally, the u update is simply

$$\mathbf{u} \leftarrow \mathbf{u} + \mathbf{D}\mathbf{x} - \mathbf{z} \tag{13}$$

This solution is implemented in the Matlab file, admm_lasso.m

The ADMM formulation for the FOH prior is also similar but has a slightly different z update scheme. For the sake of completeness, I will write out all the update steps. I do note, however, that my notational scheme is different from the Liu et al. paper and so I will write out their formulation using my notation to keep consistency. Also note that the z step comes first in this formulation and is defined per pixel.

z-step

$$z_i^{k+1} = \max\left\{\| V_i x^k + \frac{1}{\beta}u_i^k \|_2 - \frac{1}{\beta}, 0\right\}\frac{V_i x^k + \frac{1}{\beta}u_i^k}{\| V_i x^k + \frac{1}{\beta}u_i^k \|_2},$$

$$i = 1,2,..,n \tag{14}$$

x-step

$$x^{k+1} = \left(\frac{\lambda}{\beta}C^TC + V^TV\right)^{-1}\left(\frac{\lambda}{\beta}C^Tb + V^T\left(z^{k+1} - \frac{1}{\beta}u^k\right)\right) \tag{15}$$

u-step

$$u^{k+1} = u^k - \gamma\beta(z^{k+1} - Vx^{k+1}) \tag{16}$$

This implementation is carried out in the Matlab file, admm_fr.m

The last aspect of the implementation is a parameter optimization step. This simply runs each ADMM solution on an image over a set of feasible values of λ and selects the λ that maximizes the SSIM metric.

4. Analysis

As mentioned in the methodology section, there are two parts to the analysis of these methods. There is a quantitative method of performance statistics and a qualitative look at the performance on an example image.

4.1. Quantitative Analysis

There are three main measures of quantitative analysis that I will perform: PSNR, SSIM, convergence, and runtime. These statistics are calculated for each prior on each image in the image dataset.

PSNR – Peak Signal to Noise Ration

$$MSE = \frac{1}{mn}\sum_m\sum_n\left(x_{target} - x_{est}\right)^2$$

$$PSNR = 10\cdot\log_{10}\left(\frac{\max(x_{target})^2}{MSE}\right) = 10\cdot\log_{10}\left(\frac{1}{MSE}\right)$$

SSIM – Structural Similarity Index

$$\mathrm{SSIM}(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

As for convergence, a stopping condition was used:

$$\frac{\|x^{k+1} - x^k\|_2}{\|x^{k+1}\|_2} < \varepsilon$$

Additionally, this value was calculated at each iteration to measure the convergence over time.

Finally, the runtime, simply measured the elapse time between when the ADMM function is called and when the function returns.

4.2. Qualitative

The qualitative measure for these methods simply involves performing a deblur operation with each prior on the same image. This measure was done a few times with selected image conditions that illustrate the tradeoffs between the priors visually and in their performance statistics.

5. Results

The image dataset used for the tests is the the Kodak images provided here: http://r0k.us/graphics/kodak/.

These are all natural, color images which I convert to gray scale.

Performance on 24 Kodak Images

Gaussian Blur 12x12, Gaussian Noise $\sigma = 0.01$

Img	Measured		TV(anisotropic)			TV(isotropic)			Laplacian(kernel1)			Laplacian(kernel2)			Frobenius-Hessian		
Image #	PSNR(dB)	SSIM	PSNR(dB)	SSIM	Runtime(s)	PSNR(dB)	SSIM	Runtime(s)	PSNR(dB)	SSIM	Runtime(s)	PSNR(dB)	SSIM	Runtime(s)	PSNR(dB)	SSIM	Runtime(s)
1	26.5	0.57	27.34	0.64	0.39	27.34	0.64	6.46	27.49	0.64	0.2	27.49	0.64	0.2	28.08	0.66	5.3
2	25.26	0.63	26.47	0.71	0.43	26.48	0.71	5.86	26.72	0.71	0.19	26.72	0.71	0.21	27.78	0.76	4.81
3	24.57	0.56	25.72	0.63	0.46	25.71	0.63	7.55	25.91	0.64	0.21	25.91	0.64	0.22	26.63	0.67	4.92
4	17.82	0.27	18.8	0.37	0.93	18.8	0.37	16.5	18.98	0.39	0.49	18.98	0.39	0.48	19.8	0.48	4.03
5	21.22	0.39	21.91	0.44	0.33	21.92	0.44	5.87	22.07	0.45	0.17	22.07	0.45	0.17	22.76	0.51	4.12
6	22.1	0.5	22.99	0.58	0.48	22.99	0.58	8.19	23.21	0.59	0.24	23.21	0.59	0.25	24.07	0.66	4.61
7	16.02	0.25	16.9	0.34	0.63	16.9	0.34	12.01	17.06	0.36	0.36	17.06	0.36	0.38	19.01	0.55	4.99
8	22.66	0.58	23.51	0.65	0.32	23.51	0.65	6.06	23.67	0.65	0.17	23.67	0.65	0.17	25.12	0.72	4.26
9	23.31	0.55	24.34	0.62	0.39	24.34	0.62	6.85	24.56	0.62	0.19	24.56	0.62	0.19	25.59	0.68	4.49
10	19.5	0.26	20.05	0.31	0.53	20.04	0.31	8.94	20.21	0.33	0.29	20.21	0.33	0.31	21.52	0.49	4.56
11	21.89	0.44	22.81	0.51	0.6	22.81	0.51	9.38	22.98	0.52	0.26	22.98	0.52	0.26	23.75	0.57	5
12	23.8	0.58	24.85	0.64	0.26	24.86	0.65	4.61	25.16	0.65	0.14	25.16	0.65	0.14	26.86	0.7	3.57
13	18.24	0.21	18.77	0.28	0.51	18.77	0.28	9.59	18.88	0.29	0.26	18.88	0.29	0.26	19.46	0.4	3.8
14	20.95	0.36	21.79	0.44	0.51	21.79	0.44	9.53	21.96	0.45	0.26	21.96	0.45	0.27	22.5	0.51	3.55
15	22.17	0.6	23.58	0.67	0.38	23.58	0.67	6.96	23.69	0.66	0.15	23.69	0.66	0.15	25.92	0.74	4.53
16	24.91	0.51	25.75	0.57	0.39	25.75	0.57	6.41	25.89	0.58	0.17	25.89	0.58	0.17	26.42	0.61	4.7
17	22.77	0.52	23.97	0.61	0.56	23.97	0.61	10.81	24.17	0.62	0.28	24.17	0.62	0.29	25.05	0.67	5.51
18	20.65	0.36	21.43	0.42	0.64	21.43	0.42	12.19	21.57	0.43	0.35	21.57	0.43	0.36	22.08	0.47	5.68
19	20.24	0.46	20.94	0.53	0.43	20.94	0.53	7.94	21.11	0.53	0.24	21.11	0.53	0.24	22.33	0.59	4.72
20	21.46	0.63	22.69	0.7	0.28	22.71	0.7	5.2	22.79	0.69	0.14	22.79	0.69	0.15	24.68	0.75	3.65
21	20.73	0.5	21.57	0.57	0.42	21.58	0.57	8.12	21.77	0.58	0.24	21.77	0.58	0.25	22.64	0.63	4.21
22	23.17	0.44	23.88	0.49	0.41	23.88	0.49	6.62	24.05	0.5	0.18	24.05	0.5	0.19	24.76	0.55	4.62
23	24.77	0.7	25.99	0.77	0.33	25.98	0.77	5.85	26.13	0.78	0.15	26.13	0.78	0.15	27.19	0.81	4.64
24	19.66	0.37	20.49	0.43	0.54	20.49	0.43	9.44	20.65	0.45	0.28	20.65	0.45	0.28	21.48	0.51	4.48
AVG:	**21.85**	**0.468**	**22.77**	**0.538**	**0.464**	**22.77**	**0.539**	**8.20**	**22.94**	**0.546**	**0.233**	**22.94**	**0.546**	**0.239**	**23.98**	**0.612**	**4.53**

Figure 3: Each prior evaluated for 24 images

Overall, the immediate takeaways from this test is that the FOH prior does consistently better in every metric except for runtime. FOH prior consistently has a runtime that is around ten times that of the TV or Laplacian prior. This connects strongly with the initial hypothesis that FOH will perform will with natural images. Furthermore, this test illustrates that the TV and Laplacian priors are still viable options. They do have smaller PSNR and SSIM values than the FOH prior, but they still perform better than the corrupted image with a fraction of the runtime. To further consider potential tradeoffs between the priors, I also consider a visual test and convergence:

Figure 4: Reference and corrupted image, image corrupted by a 35x35 blur kernel with additive Gaussian noise with $\sigma = 0.001$.

TV Laplacian FOH

Figure 5: Reconstructions with each prior

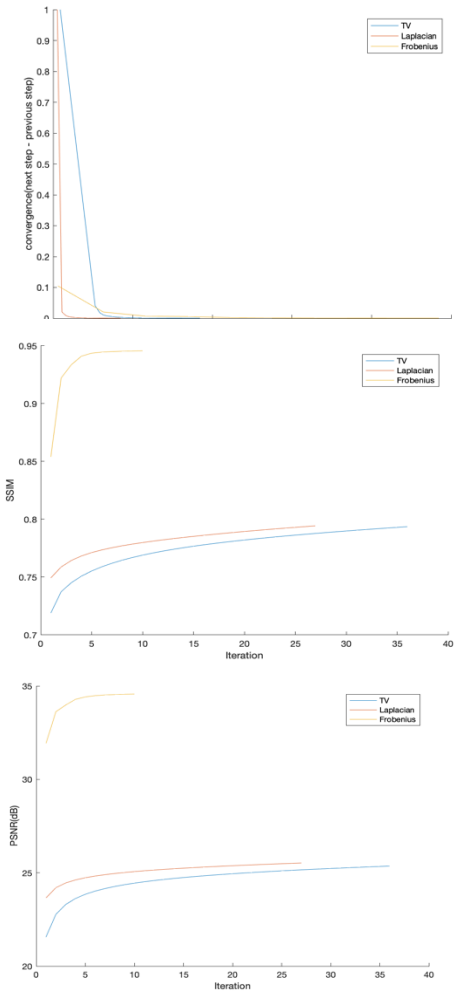


Figure 6: Graphs of prior statistics top to bottom: convergence, SSIM, PSNR

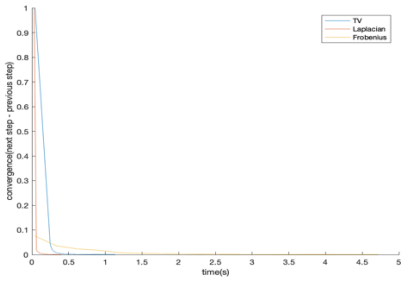
Visually speaking, the FOH clearly performs much better than the other priors for this case with large blur and low noise. Furthermore, the PSNR and SSIM are significantly higher for FOH than for the other priors.

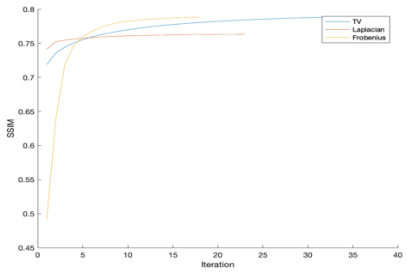
Another interesting result is that although the FOH prior starts with a lower convergence condition, it converges much slower than the TV and Laplacian priors. In this case, however, I feel that its performance outweighs the runtime tradeoff.

Figure 7: Reference and corrupted image, image corrupted by 10x10 Gaussian kernel with Gaussian noise, $\sigma = 0.05$

TV Laplacian FOH

Figure 8: Reconstructions with each prior





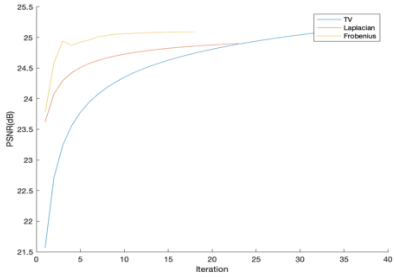


Figure 9: Graphs of prior statistics top to bottom: convergence, SSIM, PSNR

Here, the TV prior performs the best visually and again, in a fraction of the time that FOH prior operates in. The FOH does have a similar SSIM and PSNR value but visually just not seem to do as well as the TV prior. Additionally, the runtime advantage makes the TV prior much more lucrative in this case with high noise.

5.1. Conclusion

The study of the TV, Laplacian, and FOH priors relative to each other demonstrates their ability to outperform each other under different circumstances. As a result, I compile them into a single function for more robust deblurring. Putting these priors in a simpler and comparative light allows the knowledge of how to use them as a technology to become more accessible for general use. It is my hope that this paper spurs the continuation of the effort to demystify image priors with the goal of making more widely understandable and useable. Here I was limited by the time frame of the project, but others can expand on this notion with incredibly large-scale comparison of all priors of a single type that can possibly be compiled. More generalizations can also be done for novel priors as they are developed.

References

[1] S. V. Venkatakrishnan, C. A. Bouman and B. Wohlberg, "Plug-and-Play priors for model based reconstruction," *2013 IEEE Global Conference on Signal and Information Processing*, Austin, TX, 2013, pp. 945-948.

[2] Rudin, L., Osher, S., And Fatemi, E. 1992. Nonlinear total variation based noise removal algorithms. Physica D: Nonlinear Phenomena 60, 14, 259 – 268.

[3] Wetzstein, G. Image Convolution (lecture 6). EE367 Course Notes.

[4] P. Liu and L. Xiao, "Fast Hessian Frobenius Norm Based Image Restoration," *2014 Sixth International Conference on Intelligent Human-Machine Systems and Cybernetics*, Hangzhou, 2014, pp. 3-7.

[5] Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. 2001. Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends in Machine Learning 3, 1, 1–122.