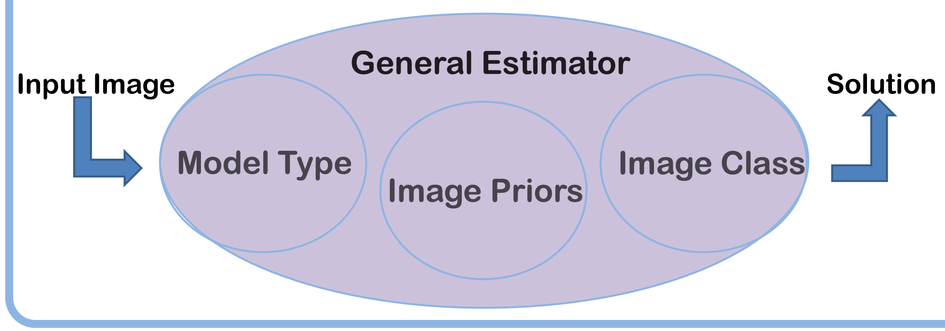
A Comparison and Compilation of Spatial Image Priors for Image Deconvolution

Amit Kohli
EE367, Stanford University

Introduction

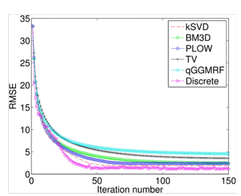
Image priors are incredibly useful tools for making ill-posed image estimation problems more tractable. It is important to thoroughly test, classify, and organize these priors for various classes of images under different conditions. This treatment of image priors is fundamental for robust and novel image estimation to become an easily usable and intuitive technology that anyone and everyone can use. I specifically implemented **spatial smoothness image priors** for **natural images** using the **ADMM** model.



Related Work

One inspiration for this project is the work of Venkatakrishnan et al in their paper 'Plug-and-Play priors for model-based reconstruction':

- Good analysis of priors against each other
- Matching of priors with inverse problem models to create generalizable implementations
- Compilation of methods into a framework that is easy to use and apply



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

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] 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.
[3] RUDIN, L., OSHER, S., AND FATEMI, E. 1992. Nonlinear total variation based noise removal algorithms. Physica D: Nonlinear Phenomena 60, 14, 259 – 268.
[4] 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.

Results

High Blur

Gaussian Blur 35x35, Gaussian Noise $\sigma = 0.001$

Measured

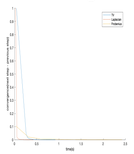
TV

Laplacian

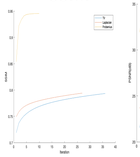
Frobenius

Frobenius clearly does the best here

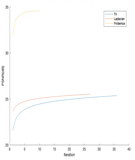
Convergence



SSIM



PSNR



High Noise

Gaussian Blur 10x10, Gaussian Noise $\sigma = 0.05$

Measured

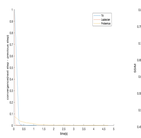
TV

Laplacian

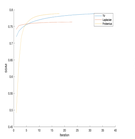
Frobenius

TV actually does the best considering convergence time

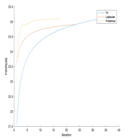
Convergence



SSIM



PSNR



Method

Image Formation Model

$$b = Cx + \eta$$

Solution format with prior

$$\text{minimize}_x \frac{1}{2} \| Cx - b \|_2^2 + \Gamma(x)$$

TV Prior

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

Laplacian

$$\Gamma(x) = |Dx|_1, D = D_{xx} + D_{yy}$$

Frobenius of Hessian

$$\Gamma(x) = |H_x x|_F, = \begin{bmatrix} D_{xx} \\ \sqrt{2} D_{xy} \\ D_{yy} \end{bmatrix} x$$

ADMM Lasso

$$\mathbf{x} = \mathbf{prox}_{f,\rho}(\mathbf{z} - \mathbf{u}) = \arg\min_{\{\mathbf{x}\}} \tfrac{1}{2} \|\mathbf{Cx} - \mathbf{b}\|_2^2 + \tfrac{\rho}{2} \|\mathbf{Dx} - \mathbf{z} + \mathbf{u}\|_2^2$$
$$\mathbf{z} = \mathbf{prox}_{g,\rho}(\mathbf{Dx} + \mathbf{u}) = \mathcal{S}_{\lambda/\rho}(\mathbf{Dx} + \mathbf{u})$$
$$\mathbf{u} = \mathbf{u} + \mathbf{Dx} - \mathbf{z}$$

ADMM Euclidean Norm

$$\mathbf{w}_i^{k+1} = \max\left\{ \|\mathbf{V}_i \mathbf{f}^k + \frac{1}{\beta}\mu_i^k\|_2 - \frac{1}{\beta}, 0 \right\} \frac{\mathbf{V}_i \mathbf{f}^k + \frac{1}{\beta}\mu_i^k}{\|\mathbf{V}_i \mathbf{f}^k + \frac{1}{\beta}\mu_i^k\|_2}$$
$$(\frac{\lambda}{\beta}\mathbf{A}^\mathbf{T}\mathbf{A} + \mathbf{V}^\mathbf{T}\mathbf{V})\mathbf{f}^{k+1} = \frac{\lambda}{\beta}\mathbf{A}^\mathbf{T}\mathbf{g} + \mathbf{V}^\mathbf{T}(\mathbf{w}^{k+1} - \frac{1}{\beta}\mu^k),$$
$$\mu^{k+1} = \mu^k - \gamma\beta(\mathbf{w}^{k+1} - \mathbf{V}\mathbf{f}^{k+1}).$$

Note:
$A = C$,
$f = x$,
$g = b$,
$w = z$,
$\mu = u$,
$|H_x x|_F = \sum_{i=1}^{n} \|\mathbf{V}_i \mathbf{f}\|_2$

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

Conclusion

Certain priors performed better with different factors:

- Blur Amount
- Noise Amount
- Image Characteristics
- Convergence Time

This project has illustrated that the compilation of priors into a general solver is useful for performing detailed comparisons between priors and is a tool that allows for better problem solving

Combining the priors into a single function proved beneficial:

- Ease of use and testing
- Adaptable to different needs
- Could have simple software integration