Trainable Algorithms for Inverse Imaging Problems

Julien Mairal

Inria Grenoble

Collaborators

Bruno
Lecouat

Jean
Ponce

Part I: Lucas-Kanade reloaded: End-to-End Super-resolution from Raw Image Bursts

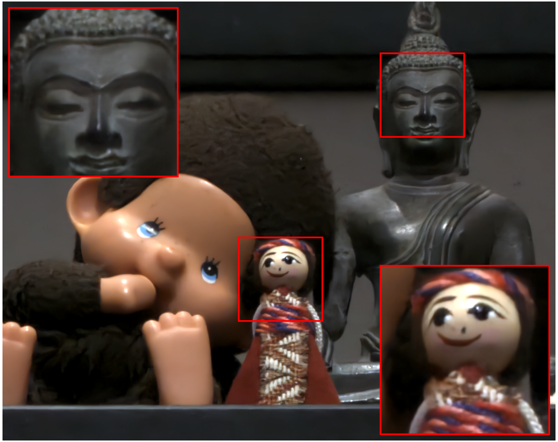
- B. Lecouat, J. Ponce, and J. Mairal. Aliasing is your Ally: End-to-End Super-resolution from Raw Image Bursts. *arXiv:2104.06191*. 2021.

A 20-megapixel innocent scene

...taken at high ISO with low exposure time

Left: high-quality jpg output of the camera ISP.

...taken at high ISO with low exposure time

Left: high-quality jpg output of the camera ISP.
Right: $\times 4$ super-resolution, after processing a burst of 30 raw images (handheld camera).

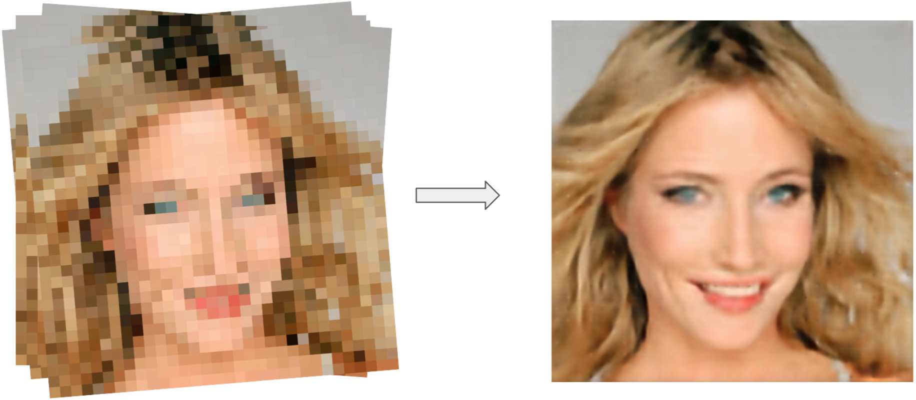
...taken at high ISO with low exposure time

Left: high-quality jpg output of the camera ISP.
Right: $\times 4$ super-resolution, after processing a burst of 30 raw images (handheld camera).

...taken at high ISO with low exposure time

Left: high-quality jpg output of the camera ISP.
Right: $\times 4$ super-resolution, after processing a burst of 30 raw images (handheld camera).

Our problem: multiframe super-resolution



low resolution frames high-resolution image

... in contrast to single-image "super-resolution"

The problem is severly **ill-posed** and the goal is to "hallucinate" high frequencies.

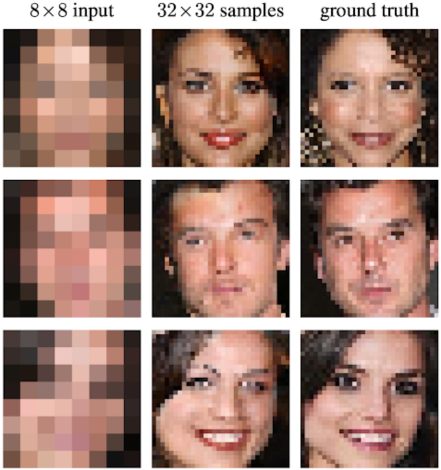


Figure: [Dahl et al., 2017]

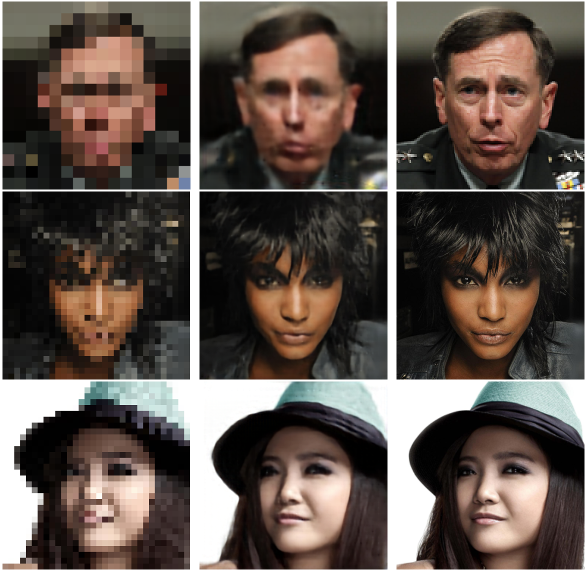
... in contrast to single-image "super-resolution"

The approach is data driven, and ...
not surprisingly...

With multiple frames: our results (in the middle)

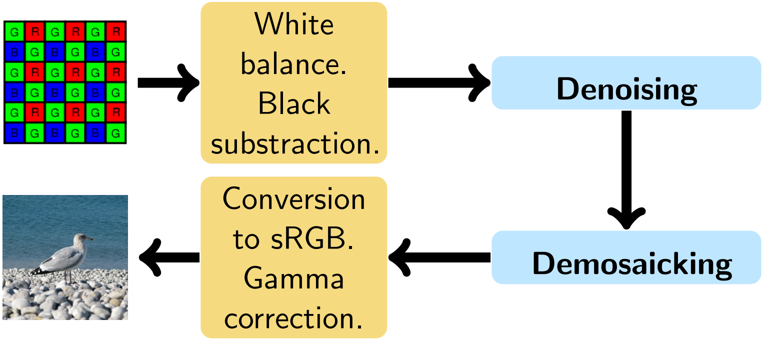
The goal is to exploit **image misalignments** to artificially increase the number of samples from the underlying signals.

20 images are generated from the ground truth with synthetic random affine movements and average pooling downsampling.



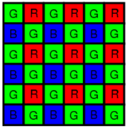
The Camera raw processing pipeline (simplified view)

How does your camera process sensor data?



The Camera raw processing pipeline (simplified view)

How does your camera process sensor data?



→ White balance. Black substraction. → **Denoising** → **Demosaicking** → Conversion to sRGB. Gamma correction. →

Idea: working with raw data is important, before the camera ISP produces irremediable damage!

With raw data, we may leverage aliasing!

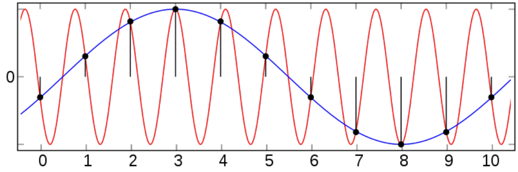


Figure: Example of aliasing: undersampled sinusoid causes confusion with a sinusoid with lower frequency. Picture from Wikipedia.

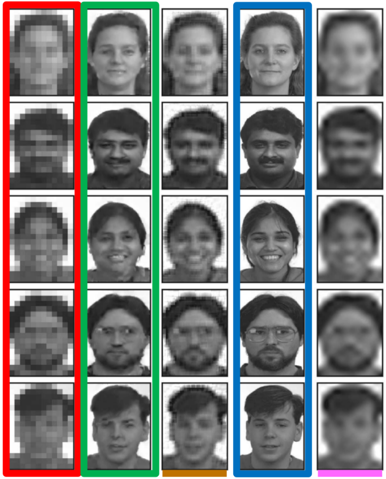
- Aliasing is usually mitigated with some optical / digital filters.
- If we analyze the aliasing patterns from multiple frames we can **recover high frequencies**.

Super-resolution from raw image bursts (with natural hand motion)

This is hard because it requires, simultaneously,

- accurately **aligning** images with subpixel accuracy.
- dealing with noisy data **(blind denoising)**.
- reconstructing color images from the Bayer pattern **(demosaicking)**.

Multiframe super resolution: prior work



(Baker and Kanade, 2002)

- LR input image (1 of 4)
- Recogstruction
- Ground-truth HR image
- (Hardie et al., 1997)
- Bicubic interpolation

x4 alignement **known exactly**

Multiframe super resolution: prior work

Handheld Multi-Frame Super-Resolution

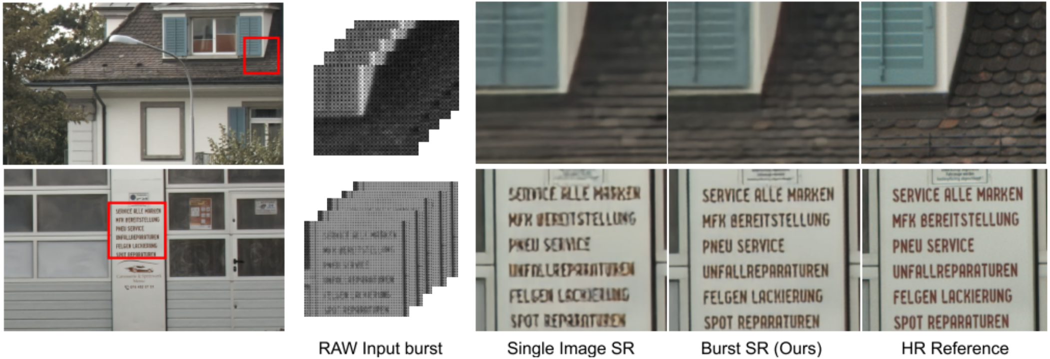
BARTLOMIEJ WRONSKI, IGNACIO GARCIA-DORADO, MANFRED ERNST, DAMIEN KELLY, MICHAEL KRAININ, CHIA-KAI LIANG, MARC LEVOY, and PEYMAN MILANFAR, Google Research

Siggraph 2020; unknown motion, raw data.

Multiframe super resolution: prior work

Deep Burst Super-Resolution

Goutam Bhat   Martin Danelljan   Luc Van Gool   Radu Timofte
Computer Vision Lab, ETH Zurich, Switzerland



Multiframe super resolution: prior work

and, among many others:

- **interpolation-based methods**: [Hardie, 2007], [Takeda et al., 2007];
- **iterative approaches**: [Irani and Peleg, 1991], [Elad and Feuer, 1997],[Farsiu et al., 2004];
- **(deep) learning-based approaches**: **[Bhat et al., 2021]**, [Molini et al., 2019], [Deudon et al., 2019];
- and also the literature on video super-resolution (typically not dealing with raw data).

Interesting for us: synthetic raw datasets from Bhat et al. [2021].

The "old" world of classical inverse problems.

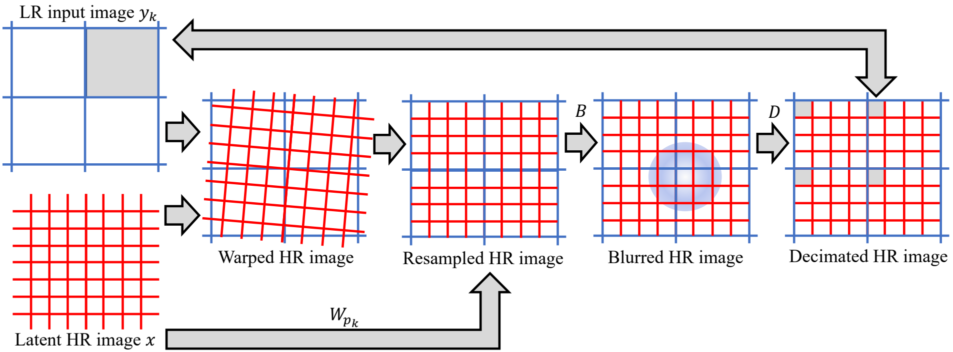


Image formation model

$$y_k = DBW_{p_k}x + \varepsilon_k.$$

The "old" world of classical inverse problems.

Inverse problem given $y_1, \ldots, y_K$

$$\min_{x, p_k} \frac{1}{K} \sum_{k=1}^{K} \| y_k - \underbrace{DBW_{p_k}}_{U_{p_k}} x \|^2 + \lambda \phi_\theta(x).$$

A natural strategy

- define an appropriate prior $\phi_\theta(x)$ for natural images.
- optimize!

The "old" world of classical inverse problems.

Inverse problem given $y_1, \ldots, y_K$

$$\min_{x, p_k} \frac{1}{K} \sum_{k=1}^{K} \| y_k - \underbrace{DBW_{p_k}}_{U_{p_k}} x \|^2 + \lambda \phi_\theta(x).$$

A natural strategy

- define an appropriate prior $\phi_\theta(x)$ for natural images.
- optimize!

Simple relaxation with "half quadratic splitting"

$$\min_{x, z, p_k} \frac{1}{K} \sum_{k=1}^{K} \| y_k - U_{p_k} z \|^2 + \frac{\mu_t}{2} \| z - x \|^2 + \lambda \phi_\theta(x).$$

The "old" world of classical inverse problems.

Simple relaxation with "half quadratic splitting" + block coordinate descent

$$\min_{x,z,p_k} \frac{1}{K} \sum_{k=1}^{K} \|y_k - U_{p_k} z\|^2 + \frac{\mu_t}{2}\|z - x\|^2 + \lambda \phi_\theta(x).$$

- minimizing with respect to p_k (parameters of an affine transformation) is performed by Gauss-Newton steps. This is the algorithm of **Lucas and Kanade [1981]**.
- minimizing with respect to x requires computing the **proximal operator** of ϕ_θ.
- minimizing w.r.t. z can be done by gradient descent steps.
- μ_t increases over the iterations.

The "old" world of classical inverse problems.

Simple relaxation with "half quadratic splitting" + block coordinate descent

$$\min_{x,z,p_k} \frac{1}{K}\sum_{k=1}^{K} \|y_k - U_{p_k} z\|^2 + \frac{\mu_t}{2}\|z - x\|^2 + \lambda \phi_\theta(x).$$

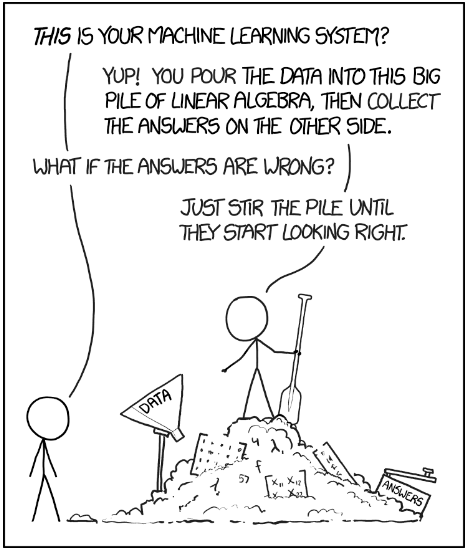
- minimizing with respect to p_k (parameters of an affine transformation) is performed by Gauss-Newton steps. This is the algorithm of **Lucas and Kanade [1981]**.
- minimizing with respect to x requires computing the **proximal operator** of ϕ_θ.
- minimizing w.r.t. z can be done by gradient descent steps.
- μ_t increases over the iterations.

Advantage: robustness and interpretability (solves what it is supposed to solve).
Drawback: designing a good image prior by hand is hard

The "new" world of deep learning models (Pic. `https://xkcd.com/`)

- a form of prior knowledge is encoded in the model architecture (*e.g.*, a convolutional neural network for images).
- ability to train model parameters θ end to end.
- state-of-the-art for many tasks (once the right model/setup is found).
- requires training data.

Advantage: task-adaptive.
Drawback: tuned to specific data distribution.



Bridging the two worlds with trainable algorithms.

Idea 1: plug-and-play priors [Venkatakrishnan et al., 2013]

Replace proximal operator

$$\arg\min_x \frac{1}{2}\|z - x\|^2 + \lambda \phi_\theta(x),$$

by a convolutional neural network $f_\theta(z)$.

Bridging the two worlds with trainable algorithms.

Idea 1: plug-and-play priors [Venkatakrishnan et al., 2013]

Replace proximal operator

$$\arg\min_{x} \frac{1}{2}\|z - x\|^2 + \lambda\phi_\theta(x),$$

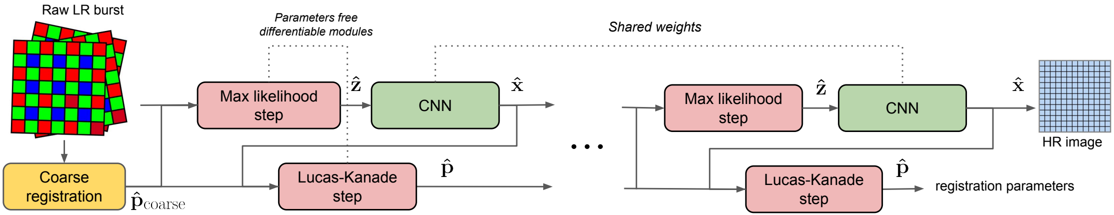
by a convolutional neural network $f_\theta(z)$.

Idea 2: unrolled optimization [Gregor and LeCun, 2010]

- Consider the previous optimization procedure with T steps, producing an estimate $\hat{x}_T(Y)$, given a burst $Y = y_1, \ldots, y_K$.
- Given a dataset of training pairs $(x_i, Y_i)_{i=1,\ldots,n}$, minimize

$$\min_{\theta} \frac{1}{n}\sum_{i=1}^{n} \|\hat{x}_T(Y_i) - x_i\|_1.$$

Schematic view of our method.



- we keep the interpretability of the classical inverse problem formulation.
- we benefit from a data-driven image prior.

Do we get the best or the worse of both worlds?

Figure: Experiment with a synthetic RGB burst of 20 images with random affine motions.

Extreme $\times 16$ super-resolution.

Figure: Experiment with a synthetic RGB burst of 20 images with random affine motions.

Experiments on real raw data - Pixel 4a.

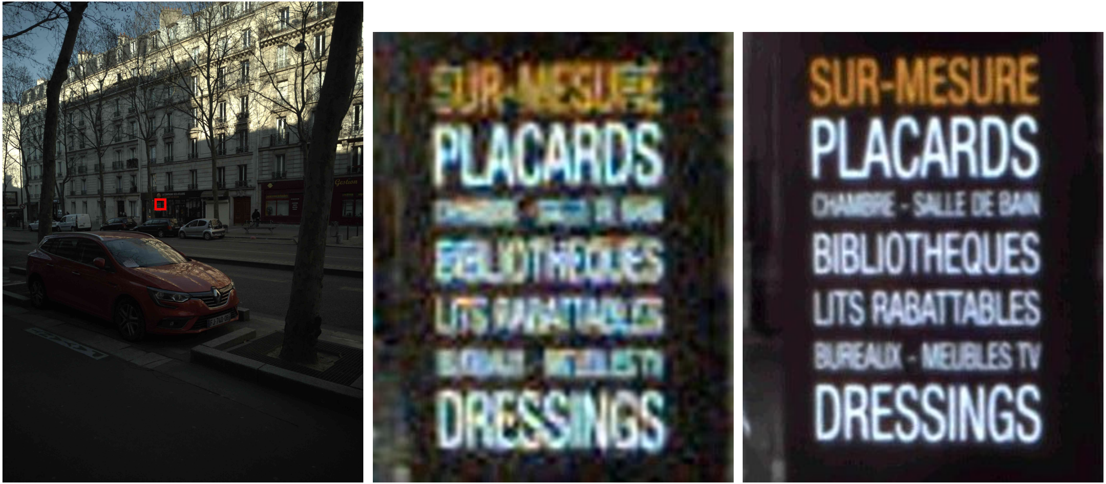


Figure: Full scene - camera ISP - Our $\times 4$ results.

Experiments on real raw data - Pixel 4a.

Figure: Full scene - camera ISP - Our $\times 4$ results.

Experiments on real raw data - Samsung S7.

Figure: Full scene - camera ISP - Our $\times 4$ results.

Experiments on real raw data - Panasonic GX9.

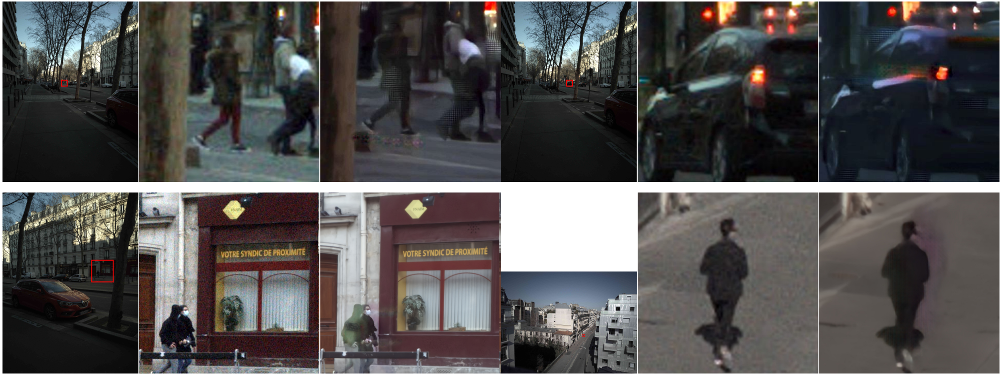
Figure: Full scene - camera ISP - Our $\times 4$ results.

Quantitative experiments.

Method	PSNR (db)	Geom (pix)	SSIM
Scores on public validation set			
ETH Bhat et al. [2021]	39.09	-	-
Ours (refine)	**41.45**	-	0.95
Scores on our own validation set to conduct the ablation study			
Multiframe + TV	34.48	-	-
Single Image	36.80	-	-
Ours (no refinements)	40.38	0.55	0.958
Ours (refinements)	**41.30**	0.32	0.963
Ours (known motion)	42.41	0.00	0.971

Table: **Results with synthetic raw image bursts** of 14 images generated from the Zurich raw to RGB dataset with synthetic affine motions. Results in average PSNR and geometrical registration error in pixels for our models.

Current issues with moving objects



full frame ISP camera Ours full frame ISP camera Ours

Figure: Misalignements artefacts due to moving objects in the scene. Our current implementation does not handle fast moving objects and then generates visual artefacts.

Conclusion

Take-home messages

- 40-years old computer vision algorithms are useful.
- aliasing is good.
- "classical" approaches are robust and intepretable and greatly benefit from deep learning principles (differentiable programming).

Future work

- microscopy and astronomical imaging where we want to recover "true" signals.
- high-quality and high-dynamic range panoramas.
- going beyond static scenes.

Part II: End-to-End Sparse Coding Models

- B. Lecouat, J. Ponce, and J. Mairal. Fully Trainable and Interpretable Non-Local Sparse Models for Image Restoration. *(ECCV)*. 2020.
- B. Lecouat, J. Ponce, and J. Mairal. Designing and Learning Trainable Priors with Non-Cooperative Games. *(NeurIPS)*. 2020.

Image denoising: classical image models

$$\underbrace{y}_{\text{measurements}} = \underbrace{x_0}_{\text{original image}} + \underbrace{\varepsilon}_{\text{noise}}.$$

Energy minimization problem - MAP estimation

$$E(x) = \underbrace{\frac{1}{2}\|y - x\|_2^2}_{\text{relation to measurements}} + \underbrace{\psi(x)}_{\text{image model}}.$$

Some classical priors

- Smoothness $\lambda\|\mathcal{L}x\|_2^2$;
- total variation $\lambda\|\nabla x\|_1^2$ [Rudin et al., 1992];
- Markov random fields [Zhu and Mumford, 1997];
- wavelet sparsity $\lambda\|D^\top x\|_1$.

Image denoising

The method of Elad and Aharon [2006]

Given a **fixed** dictionary D, a patch y_i (*e.g.*, 8×8) is denoised as follows:

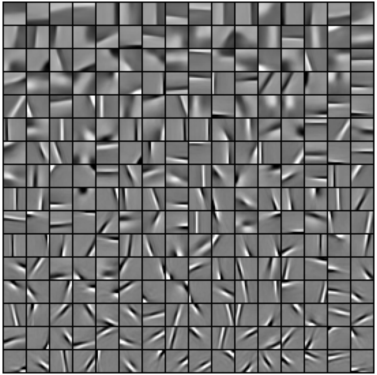
1. center y_i,
$$y_i^c \triangleq y_i - \mu_i \mathbf{1}_m \quad \text{with} \quad \mu_i \triangleq \frac{1}{n}\mathbf{1}_m^\top y_i;$$
2. find a sparse linear combination of dictionary elements that approximates y_i^c up to the noise level:
$$\min_{\alpha_i \in \mathbb{R}^p} \|\alpha_i\|_0 \quad \text{s.t.} \quad \|y_i^c - \mathbf{D}\alpha_i\|_2^2 \leq \varepsilon, \tag{1}$$
where ε is proportional to the noise variance σ^2;
3. add back the mean component to obtain the clean estimate $\hat{x}_i$:
$$\hat{x}_i \triangleq D\alpha_i^\star + \mu_i \mathbf{1}_m,$$

Image denoising

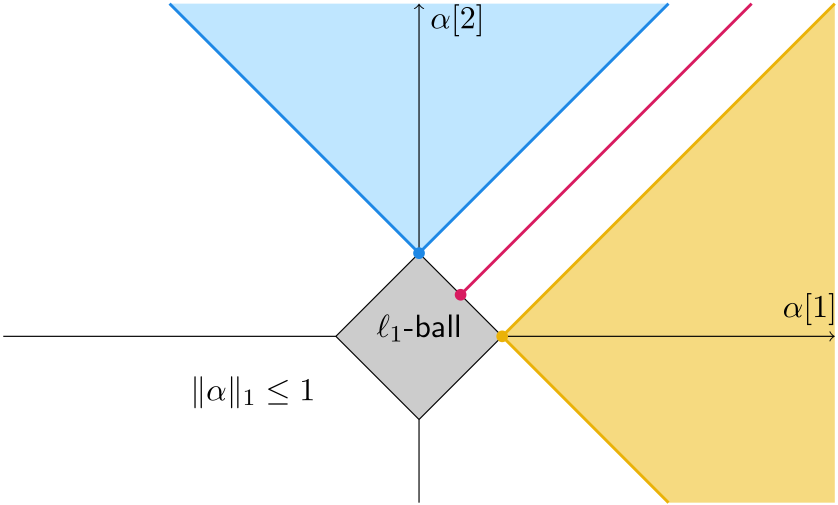
The method of Elad and Aharon [2006]

An **adaptive** approach

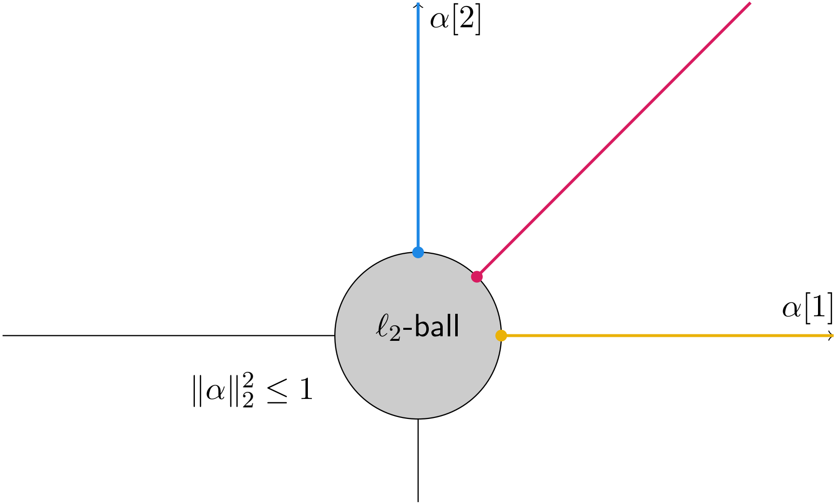
1. extract all overlapping $\sqrt{m} \times \sqrt{m}$ patches y_i.
2. **dictionary learning**: learn D on the set of centered noisy patches $[y_1^c, \ldots, y_n^c]$.
3. **final reconstruction**: find an estimate $\hat{x}_i$ for every patch using the approach of the previous slide;
4. **patch averaging**: $\hat{x} = \frac{1}{m} \sum_{i=1}^{n} \mathbf{R}_i^\top \hat{x}_i$.



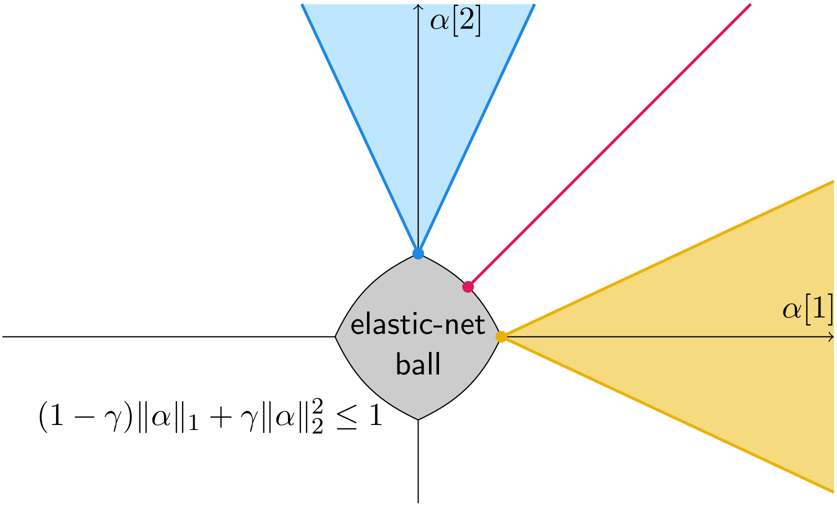
Parenthesis on sparsity-inducing penalties



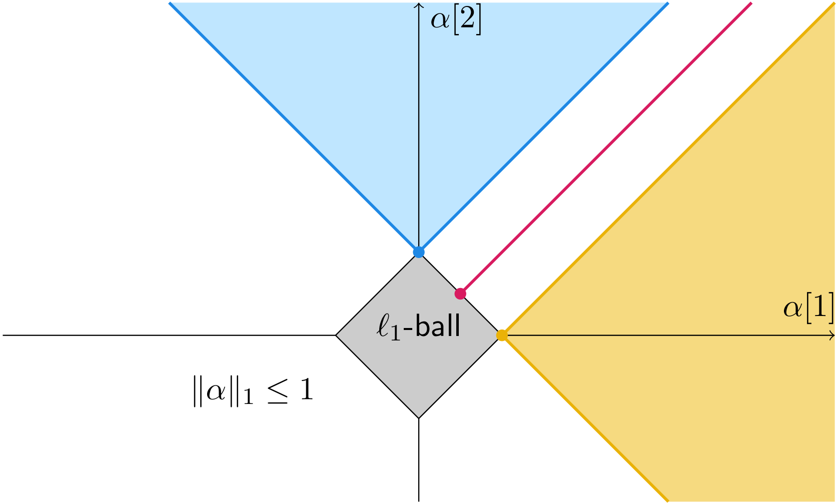
Parenthesis on sparsity-inducing penalties



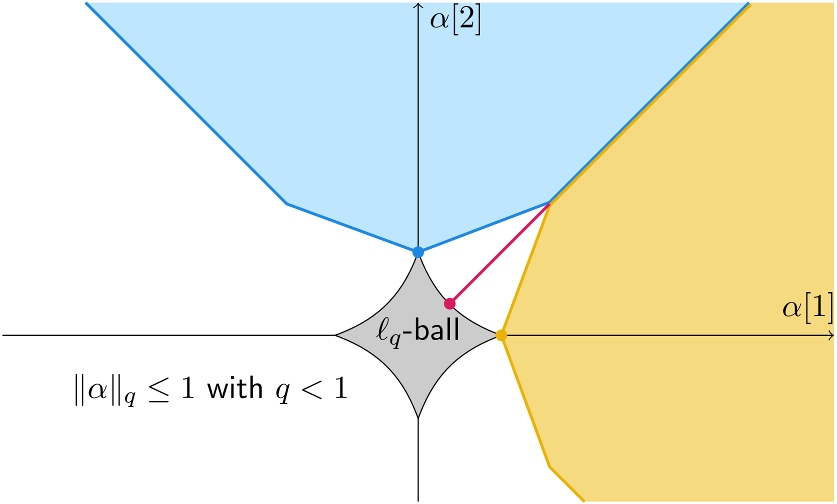
Parenthesis on sparsity-inducing penalties



Parenthesis on sparsity-inducing penalties



Parenthesis on sparsity-inducing penalties



Other patch modeling approaches

Non-local means and non-parametric approaches

Image pixels are well explained by a Nadaraya-Watson estimator:

$$\hat{x}[i] = \sum_{j=1}^{n} \frac{K_h(y_i - y_j)}{\sum_{l=1}^{n} K_h(y_i - y_l)} y[j], \tag{2}$$

with successful application to

- texture synthesis: [Efros and Leung, 1999]
- image denoising **(Non-local means)**: [Buades et al., 2005]
- image demosaicking: [Buades et al., 2009].

Other patch modeling approaches

BM3D

a state-of-the-art image denoising approach [Dabov et al., 2007]:

- **block matching**: for each patch, find similar ones in the image;
- **3D wavelet filtering**: denoise blocks of patches with 3D-DCT;
- **patch averaging**: average estimates of overlapping patches;
- **second step with Wiener filtering**: use the first estimate to perform again and improve the previous steps.

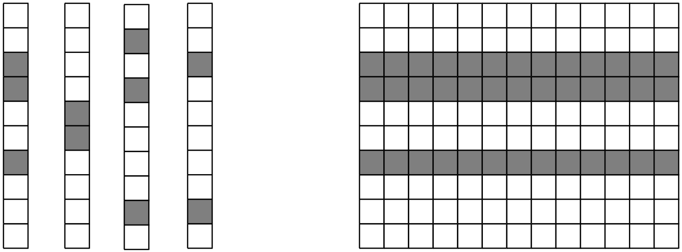
Further refined by Dabov et al. [2009] with shape-adaptive patches and PCA filtering.

Other patch modeling approaches

Non-local sparse models [Mairal et al., 2009]

Combine the non-local means principle with dictionary learning.

The main idea is that **similar patches should admit similar decompositions** by using group sparsity:



The approach uses group sparse coding and patch averaging.

How to derive a trainable algorithm from sparse coding principles?

Consider the Lasso problem

$$\min_{\alpha \in \mathbb{R}^p} \frac{1}{2}\|x - D\alpha\|^2 + \lambda\|\alpha\|_1$$

A classical algorithm to solve the optimization problem is the proximal gradient descent method

$$\alpha_t \leftarrow \mathsf{prox}_{\lambda\|.\|_1}\left[\alpha_{t-1} - \eta D^\top(D\alpha_{t-1} - x)\right]$$

This motivates the LISTA approach consisting of unrolling T steps of

$$\alpha_t \leftarrow \mathsf{prox}_{\lambda\|.\|_1}\left[\alpha_{t-1} + C^\top(D\alpha_{t-1} - x)\right],$$

and see the resuling iterate $\alpha_T(x)$ as a parametric function of D and C.

How to derive a trainable algorithm from sparse coding principles?

Consider the Lasso problem

$$\min_{\alpha \in \mathbb{R}^p} \frac{1}{2}\|x - D\alpha\|^2 + \lambda\|\alpha\|_1$$

A classical algorithm to solve the optimization problem is the proximal gradient descent method

$$\alpha_t \leftarrow \mathsf{prox}_{\lambda\|.\|_1}\left[\alpha_{t-1} - \eta D^\top(D\alpha_{t-1} - x)\right]$$

This motivates the LISTA approach consisting of unrolling T steps of

$$\alpha_t \leftarrow \mathsf{prox}_{\lambda\|.\|_1}\left[\alpha_{t-1} + C^\top(D\alpha_{t-1} - x)\right],$$

and see the resuling iterate $\alpha_T(x)$ as a parametric function of D and C.

Remark: One steps performs an affine transformation of α_{t-1} + prox.

How to derive a trainable algorithm from sparse coding principles?

Consider the Lasso problem

$$\min_{\alpha \in \mathbb{R}^p} \frac{1}{2}\|x - D\alpha\|^2 + \lambda\|\alpha\|_1$$

A classical algorithm to solve the optimization problem is the proximal gradient descent method

$$\alpha_t \leftarrow \mathsf{prox}_{\lambda\|.\|_1}\left[\alpha_{t-1} - \eta D^\top(D\alpha_{t-1} - x)\right]$$

This motivates the LISTA approach consisting of unrolling T steps of

$$\alpha_t \leftarrow \mathsf{prox}_{\lambda\|.\|_1}\left[\alpha_{t-1} + C^\top(D\alpha_{t-1} - x)\right],$$

and see the resuling iterate $\alpha_T(x)$ as a parametric function of D and C.

Remark: One steps performs an affine transformation of α_{t-1} + prox.

Remark 2: prox$_{\lambda\|.\|_1}$ $[u] = \text{sign}(u) * RELU(|u| - \lambda)$

End-to-end sparse coding models

Denoising with the ℓ_1-norm

Then, we can "unroll" the patch-based sparse coding denoising approach and learn the matrices D and C in a supervised fashion, given training pairs of noisy/clean images, as done by Simon and Elad [2019].

Denoising with non-local sparse models (our work)

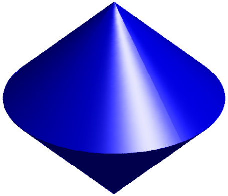
- deal with the **proximal operator of the Group Lasso penalty**.
- **take into account self-similarities** via similarity matrix Σ.
- end-to-end learning with unrolled optimization.

Group Lasso and mixed norms

[Turlach et al., 2005, Yuan and Lin, 2006, Zhao et al., 2009]
[Grandvalet and Canu, 1999, Bakin, 1999]

$$\text{the } \ell_1/\ell_q\text{-norm :} \qquad \psi(\alpha) = \sum_{g \in \mathcal{G}} \|\alpha[g]\|_2.$$

- $\mathcal{G}$ is a **partition** of $\{1, \ldots, p\}$;
- $q = 2$ or $q = \infty$ in practice;
- can be interpreted as the ℓ_1-norm of $[\|\alpha[g]\|_2]_{g \in \mathcal{G}}$.



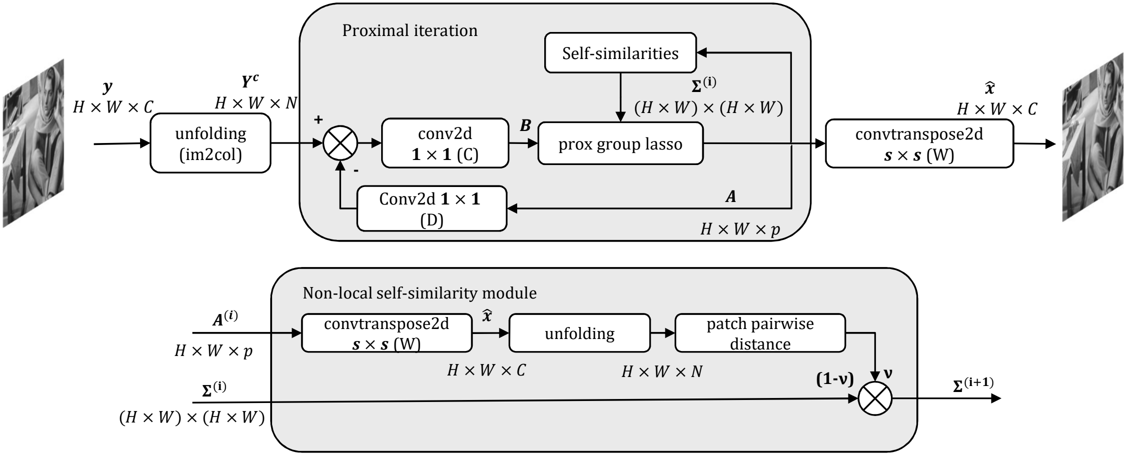
$$\psi(\alpha) = \|\alpha[\{1,2\}]\|_2 + |\alpha[3]|.$$

End-to-end sparse coding models

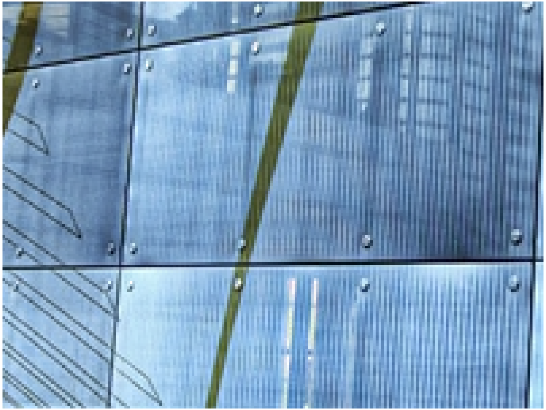
Algorithm 1 Pseudo code for the inference model of GroupSC.

```
Extract patches $Y = [y_1, \ldots, y_n]$ and center them;
Initialize the codes $\alpha_i$ to $0$;
Initialize image estimate $\hat{x}$ to the noisy input $y$;
Initialize pairwise similarities $\Sigma$ between patches of $\hat{x}$;
for $k = 1, 2, \ldots K$ do
    Compute pairwise patch similarities $\hat{\Sigma}$ on $\hat{x}$;
    Update $\Sigma \leftarrow (1 - \nu)\Sigma + \nu\hat{\Sigma}$;
    for $i = 1, 2, \ldots, N$ in parallel do
        $\alpha_i \leftarrow \text{prox}_{\Sigma,\Lambda_k} \left[\alpha_i + C^\top (y_i^c - D\alpha_i)\right]$;
    end for
    Update the denoised image $\hat{x}$ by averaging;
end for
```

For the young generation



Demosaicking. Sparse coding vs. non-local sparse coding

Denoising. Sparse coding vs. non-local sparse coding

JPEG Deblocking. Sparse coding vs. non-local sparse coding

Interesting conclusion: parameter-efficient models

Table: **Demosaicking.** Training on CBSD400 unless a larger dataset is specified between parenthesis. Performance is measured in terms of average PSNR.

Method	Trainable	Params	Kodak24	BSD68	Urban100
LSSC	✗	-	41.39	40.44	36.63
IRCNN (BSD400+Waterloo)		-	40.54	39.9	36.64
Kokinos (MIT dataset)		380k	41.5	-	-
MMNet (MIT dataset)		380k	42.0	-	-
RNAN		8.96M	**42.86**	42.61	-
SC (ours)		119k	42.34	41.88	37.50
CSR (ours)		119k	42.25	-	-
GroupSC (ours)		119k	42.71	**42.91**	**38.21**

Interesting conclusion: parameter-efficient models

Table: **Grayscale Denoising** on BSD68, training on BSD400 for all methods. Performance is measured in terms of average PSNR.

Method	Trainable	Params	Noise Level (σ) 5	15	25	50
BM3D	✗	-	37.57	31.07	28.57	25.62
LSSC	✗	-	37.70	31.28	28.71	25.72
BM3D PCA	✗	-	37.77	31.38	28.82	25.80
TNRD		-	-	31.42	28.92	25.97
CSCnet		62k	37.84	31.57	29.11	26.24
LKSVD		45K	-	31.54	29.07	26.13
FFDNet		486k	-	31.63	29.19	26.29
DnCNN		556k	37.68	<u>31.73</u>	29.22	26.23
NLRN		330k	<u>37.92</u>	**31.88**	**29.41**	**26.47**
SC (baseline)		68k	37.84	31.46	28.90	25.84
GroupSC (ours)		68k	**37.95**	31.71	29.20	26.17

Interesting conclusion: leveraging interpretability

Table: **Blind denoising** on CBSD68, training on CBSD400. Performance is measured in terms of average PSNR. Best is in bold, second is underlined.

Noise level	CBM3D -	CDnCNN-B 666k	CUNet 93k	CUNLnet 93k	SC (ours) 115k	GroupSC (ours) 115k
5	40.24	40.11	40.31	<u>40.39</u>	40.30	**40.43**
10	35.88	36.11	36.08	<u>36.20</u>	36.07	**36.29**
15	33.49	33.88	33.78	<u>33.90</u>	33.72	**34.01**
20	31.88	<u>32.36</u>	32.21	32.34	32.11	**32.41**
25	30.68	<u>31.22</u>	31.03	31.17	30.91	**31.25**

- learn common D, C parameters for different noise levels.
- learn noise-specific regularization parameters λ

$$\min_{\alpha \in \mathbb{R}^p} \frac{1}{2}\|x - D\alpha\|^2 + \lambda\|\alpha\|_1.$$

Conclusion

On trainable algorithms

- this is a **hybrid** point of view between deep learning black boxes and **classical inverse problem formulations**.
- This is a natural way to encode **a priori knowledge** in the model and obtain smaller models.
- **Intepretability is useful!**

Caveats

- unrolled optimization is often unstable and requires heuristics for training.
- not much theory (leading to exciting new challenges).

References I

S. Bakin. *Adaptive regression and model selection in data mining problems*. PhD thesis, 1999.

Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Deep burst super-resolution. *arXiv preprint arXiv:2101.10997*, 2021.

A. Buades, B. Coll, and J.-M. Morel. A review of image denoising algorithms, with a new one. *SIAM Journal on Multiscale Modeling and Simulation*, 4(2):490–530, 2005.

A. Buades, B. Coll, J.-M. Morel, and C. Sbert. Self-similarity driven color demosaicking. *IEEE Transactions on Image Processing*, 18(6):1192–1202, 2009.

K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3D transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095, 2007.

K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. BM3D image denoising with shape-adaptive principal component analysis. In *SPARS'09-Signal Processing with Adaptive Sparse Structured Representations*, 2009.

Ryan Dahl, Mohammad Norouzi, and Jonathon Shlens. Pixel recursive super resolution. In *Proceedings of the international conference on computer vision (ICCV)*, 2017.

References II

Michel Deudon, Alfredo Kalaitzis, Md Rifat Arefin, Israel Goytom, Zhichao Lin, Kris Sankaran, Vincent Michalski, Samira E Kahou, Julien Cornebise, and Yoshua Bengio. Highres-net: Multi-frame super-resolution by recursive fusion. 2019.

A.A. Efros and T.K. Leung. Texture synthesis by non-parametric sampling. In *Proc. International Conference on Computer Vision (ICCV)*, 1999.

M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 15(12):3736–3745, 2006.

Michael Elad and Arie Feuer. Restoration of a single superresolution image from several blurred, noisy, and undersampled measured images. *IEEE transactions on image processing*, 6(12): 1646–1658, 1997.

Sina Farsiu, M Dirk Robinson, Michael Elad, and Peyman Milanfar. Fast and robust multiframe super resolution. *IEEE transactions on image processing*, 13(10):1327–1344, 2004.

Y. Grandvalet and S. Canu. Outcomes of the equivalence of adaptive ridge with least absolute shrinkage. In *Advances in Neural Information Processing Systems (NIPS)*, 1999.

References III

Karol Gregor and Yann LeCun. Learning fast approximations of sparse coding. In *Proc. International Conference on Machine Learning (ICML)*, 2010.

Russell Hardie. A fast image super-resolution algorithm using an adaptive wiener filter. *IEEE Transactions on Image Processing*, 16(12):2953–2964, 2007.

Michal Irani and Shmuel Peleg. Improving resolution by image registration. *CVGIP: Graphical models and image processing*, 53(3):231–239, 1991.

Bruce D Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of Imaging Understanding Workshop*, 1981.

J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Non-local sparse models for image restoration. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2009.

Andrea Bordone Molini, Diego Valsesia, Giulia Fracastoro, and Enrico Magli. Deepsum: Deep neural network for super-resolution of unregistered multitemporal images. *IEEE Transactions on Geoscience and Remote Sensing*, 58(5):3644–3656, 2019.

L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1-4):259–268, 1992.

References IV

Dror Simon and Michael Elad. Rethinking the CSC model for natural images. *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Hiroyuki Takeda, Sina Farsiu, and Peyman Milanfar. Kernel regression for image processing and reconstruction. *IEEE Transactions on image processing*, 16(2):349–366, 2007.

B.A. Turlach, W.N. Venables, and S.J. Wright. Simultaneous variable selection. *Technometrics*, 47 (3):349, 2005.

Singanallur V Venkatakrishnan, Charles A Bouman, and Brendt Wohlberg. Plug-and-play priors for model based reconstruction. In *IEEE Global Conference on Signal and Information Processing*, pages 945–948. IEEE, 2013.

M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society Series B*, 68:49–67, 2006.

P. Zhao, G. Rocha, and B. Yu. The composite absolute penalties family for grouped and hierarchical variable selection. *Annals of Statistics*, 37(6A):3468–3497, 2009.

S. C. Zhu and D. Mumford. Prior learning and gibbs reaction-diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(11):1236–1250, 1997.