Fast L0-based Sparse Signal Recovery

Nick Kingsbury and Yingsong Zhang

UNIVERSITY OF CAMBRIDGE

Signal Processing Group
Department of Engineering

ACVT seminar, Adelaide – May 2011

Outline

1. Introduction
2. L0 reweighted-L2 Minimization
   - IRLS
   - Basic Model
   - Basic Algorithm
   - Fast Algorithm $\mathrm{L_0RL_2}$
3. Continuation Strategy
   - Geometry of the penalty function
   - $\mathrm{L_0RL_2}$ with continuation
4. Numerical Results
5. Super-resolution
6. Experimental results
7. Conclusion

Introduction

The L_0 minimisation problem

$$\text{Find } \mathbf{x} \text{ which gives} \quad \min \|\mathbf{x}\|_0 \quad \text{subject to} \quad \|\mathbf{y} - \Phi\mathbf{x}\|_2 \leq \xi \qquad (1)$$

where Φ is known and of size $M \times N$ with $M < N$

- L_0: NP-hard combinatorial search for exact solution
- L_1: convex problem, but poor computation speed for large problems; greedy algorithms can be used, e.g. CoSaMP, NESTA.
- L_p: iterative reweighted techniques, e.g. IRL1, IRLS
- L_0: greedy algorithms usually used, e.g. IHT, ℓ_0-AP, SL0

We propose a highly efficient form of IRLS which approximates L_0 minimisation, allows $N \sim 10^7$ or 10^8, and has good reconstruction performance.

Iterative reweighted least-squares minimization (IRLS)

$$\min \mathbf{x}^T \mathbf{S} \mathbf{x} \quad \text{subject to} \quad \|\mathbf{y} - \Phi \mathbf{x}\|_2 = 0$$

$\mathbf{S}$ is a diagonal weight matrix. The solution of each step is the solution of

$$\min \|\mathbf{v}\|^2 \quad \text{subject to } \|\mathbf{y} - \Phi \mathbf{x}\|_2 = 0 \quad \text{and} \quad \mathbf{x} = \mathbf{S}^{-\frac{1}{2}} \mathbf{v}$$

which is unique and the reweighted iterative rule is:

$$\mathbf{x} = \mathbf{S}^{-\frac{1}{2}} (\Phi \mathbf{S}^{-\frac{1}{2}})^{\dagger} \mathbf{y}, \tag{2}$$

where † denotes the pseudo inverse (i.e. $\mathbf{H}^{\dagger} = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T$).

The inverse of $(\mathbf{H}^T \mathbf{H})$ is difficult for large problems.

Iterative reweighted least squares minimizations (IRLS)

Gorodnitsky & Rao (1997) $S_j = \frac{1}{\left(|x_j|^2\right)^{(1-\frac{p}{2})}}, \quad \text{for } 0 \leq p \leq 1\,.$

Chartrand & Yin (2008) $S_j = \frac{1}{\left(|x_j|^2+\epsilon^2\right)^{(1-\frac{p}{2})}}, \quad \text{for } 0 \leq p \leq 1$

Daubechies et al (2009)

- Theoretical analysis shows that local convergence rate of the algorithm is superlinear; smaller p values result in faster convergence rate.
- Experimentally shows that L_p norm with $p \to 0$ can help to achieve a higher success rate in exact signal recovery.

However, IRLS in this form strictly only models noise-free observations.

Basic Model – Gaussian measurement noise

Consider the noisy system

$$\mathbf{y} = \Phi\mathbf{x} + \mathbf{n} \tag{3}$$

where we assume :

$$\Phi \text{ is a known } M \times N \text{ matrix}, \qquad \mathbf{n} \sim \mathcal{N}(0, \nu^2) \text{ is noise}$$

and the prior of $\mathbf{x}$ is a zero-mean scaled adaptive Gaussian model such that

$$p(\mathbf{x}) \propto \sqrt{|\mathbf{S}|}\exp(-\tfrac{1}{2}\mathbf{x}^T\mathbf{S}\mathbf{x}),$$

where $\mathbf{S}$ is a diagonal matrix, whose j^{th} diagonal entry $S_j = 1/\sigma_j^2$.

This is well suited for modeling wavelet coefs.

Basic Model – Prior for σ_j

We further assume an independent prior $\propto \exp(-\frac{1}{2}\epsilon^2/\sigma_j^2)$ for each σ_j.

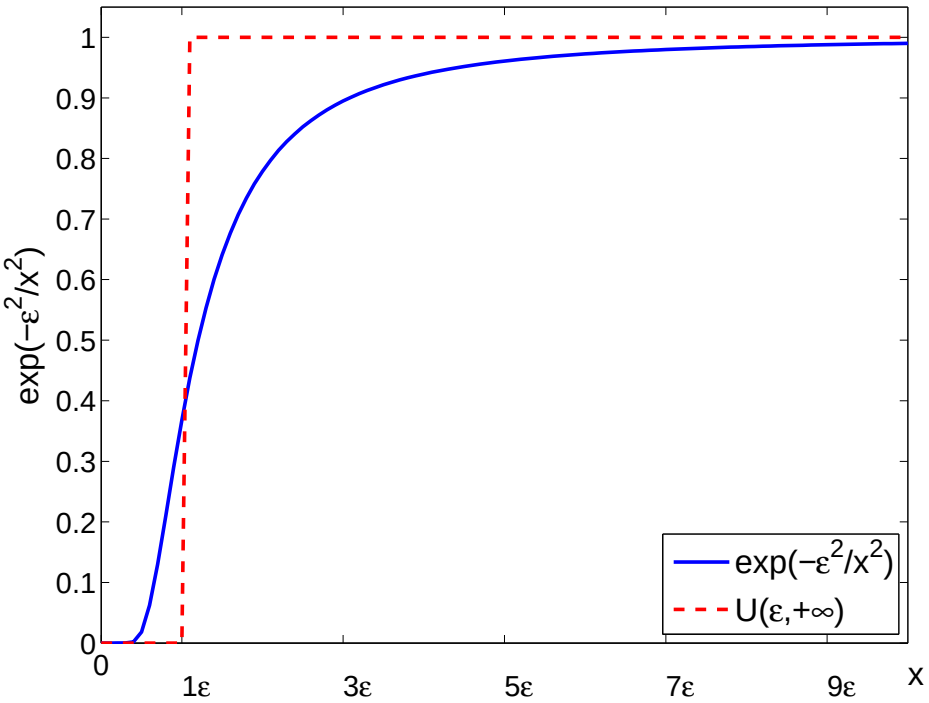


Figure: Illustration of the prior for σ_j.

This prior can be regarded as an approximation to the lower bounded uniform prior $U(\epsilon, +\infty)$. It is approximately uniformly distributed in the region $(3\epsilon, +\infty)$. Meanwhile it tends to 0 as σ_j approaches 0 so as to prevent σ^2 getting too small and hence avoid numerical instability in the model.

Log MAP function and basic algorithm

The prior on σ_j gives the following negative log MAP function:

Negative log MAP function

$$J(\mathbf{x},\mathbf{S}) = \nu^2 \left(\mathbf{x}^T\mathbf{S}\mathbf{x} - \ln|\mathbf{S}| + \epsilon^2 \sum_j S_j \right) + \|\mathbf{y} - \Phi\mathbf{x}\|^2 \qquad (4)$$

Minimizing $J(\mathbf{x},\mathbf{S})$ results in the following iteration rules:

Basic algorithm

$$\left.\begin{aligned} \mathbf{x} &= \arg\min_{\mathbf{x}} J(\mathbf{x},\mathbf{S}) = (\Phi^T\Phi + \nu^2\mathbf{S})^{-1}\Phi^T\mathbf{y} \\ \sigma_j^2 &= \arg\min_{\sigma_j^2} J(\mathbf{x},\mathbf{S}) = |x_j|^2 + \epsilon^2 \\ S_j &= \frac{1}{\sigma_j^2} = \frac{1}{|x_j|^2 + \epsilon^2} \qquad \forall\, j \end{aligned}\right\} \qquad (5)$$

Fast Algorithm L_0RL_2 – to avoid $(\Phi^T\Phi + \nu^2\mathbf{S})^{-1}$

For wavelet-like signal spaces, we introduce the vector $\boldsymbol{\alpha} = [\alpha_1 \dots \alpha_K]$ and the diagonal operator Λ_α that multiplies the k^{th} subspace/subband by α_k:

$$(\Lambda_\alpha \mathbf{x})_k = \alpha_k \mathbf{x}_k \qquad \text{for} \quad k = 1 \cdots K$$

where $\mathbf{x}_k$ is a masked version of $\mathbf{x}$ with non-zeros only in the subband k, and where α_k may be optimized independently for each subband to be the minimum α_k such that

$$\alpha_k \mathbf{x}_k^T \mathbf{x}_k \geq \|\Phi \mathbf{x}_k\|^2 \qquad \text{for any } k \text{ and } \mathbf{x}.$$

Then using majorisation minimization (MM) [3], we have the following

new auxiliary function:

$$J_\alpha(\mathbf{x}, \mathbf{S}, \mathbf{z}) = J(\mathbf{x}, \mathbf{S}) + (\mathbf{x} - \mathbf{z})^T \Lambda_\alpha (\mathbf{x} - \mathbf{z}) - \|\Phi(\mathbf{x} - \mathbf{z})\|^2 \qquad (6)$$

Fast Algorithm L_0RL_2

The new auxiliary function eliminates the difficult $\mathbf{x}^T\Phi^T\Phi\mathbf{x}$ term from $J(x, S)$ and results in the following iteration rules:

L_0RL_2

$$\left.\begin{aligned}\mathbf{x}_{n+1} &= (\Lambda_\alpha + \nu^2\mathbf{S}_n)^{-1}\left[(\Lambda_\alpha - \Phi^T\Phi)\mathbf{z}_n + \Phi^T\mathbf{y}\right] \\ \mathbf{z}_{n+1} &= \arg\min_{\mathbf{z}} J_\alpha(\mathbf{x}_{n+1}, \mathbf{S}_n, \mathbf{z}) = \mathbf{x}_{n+1}\end{aligned}\right\} \quad (7a)$$

$$\mathbf{S}_{n+1} = \mathrm{diag}\left(\left[|x_{j,n+1}|^2 + \epsilon^2\right]^{-1}_{j=1,\cdots,N}\right) \quad (7b)$$

Note that $(\Lambda_\alpha + \nu^2\mathbf{S}_n)$ is now a diagonal matrix and hence is easy to invert!

$$J(\mathbf{x},\mathbf{S}) \;=\; \nu^2\left(\mathbf{x}^T\mathbf{S}\mathbf{x} - \ln|\mathbf{S}| + \epsilon^2\sum\nolimits_j S_j\right) + \|\mathbf{y}-\Phi\mathbf{x}\|^2 \qquad (4)$$

$$\left.\begin{aligned}\mathbf{x}_{n+1} &= (\Lambda_\alpha + \nu^2\mathbf{S}_n)^{-1}\left[(\Lambda_\alpha - \Phi^T\Phi)\mathbf{z}_n + \Phi^T\mathbf{y}\right]\\ \mathbf{z}_{n+1} &= \arg\min_{\mathbf{z}} J_\alpha(\mathbf{x}_{n+1},\mathbf{S}_n,\mathbf{z}) = \mathbf{x}_{n+1}\end{aligned}\right\} \qquad (7a)$$

$$\mathbf{S}_{n+1} = \mathrm{diag}\left(\left[|x_{j,n+1}|^2 + \epsilon^2\right]^{-1}_{j=1,\cdots,N}\right) \qquad (7b)$$

Substituting eq(7b) into log MAP eq(4) gives

cost function

$$J(\mathbf{x},\mathbf{S}) = \nu^2\left(\mathrm{const} + \sum\nolimits_j \ln(x_j^2+\epsilon^2)/\epsilon\right) + \|\mathbf{y}-\Phi\mathbf{x}\|^2 \qquad (9)$$

$$J(\mathbf{x},\mathbf{S}) = \nu^2 \left(\mathbf{x}^T \mathbf{S} \mathbf{x} - \ln |\mathbf{S}| + \epsilon^2 \sum\nolimits_j S_j \right) + \|\mathbf{y} - \Phi\mathbf{x}\|^2 \qquad (4)$$

$$\left.\begin{aligned} \mathbf{x}_{n+1} &= (\Lambda_\alpha + \nu^2 \mathbf{S}_n)^{-1} \left[(\Lambda_\alpha - \Phi^T\Phi)\mathbf{z}_n + \Phi^T\mathbf{y} \right] \\ \mathbf{z}_{n+1} &= \arg\min_{\mathbf{z}} J_\alpha(\mathbf{x}_{n+1}, \mathbf{S}_n, \mathbf{z}) = \mathbf{x}_{n+1} \end{aligned}\right\} \qquad (7a)$$

$$\mathbf{S}_{n+1} = \mathrm{diag}\left(\left[|x_{j,n+1}|^2 + \epsilon^2 \right]_{j=1,\cdots,N}^{-1} \right) \qquad (7b)$$

Substituting eq(7b) into log MAP eq(4) gives

cost function

$$J(\mathbf{x},\mathbf{S}) = \nu^2 \left(\mathrm{const} + \sum\nolimits_j \ln(x_j^2 + \epsilon^2)/\epsilon \right) + \|\mathbf{y} - \Phi\mathbf{x}\|^2 \qquad (9)$$

Geometry of the log-sum penalty function $f_{\ln,\epsilon}$

$$f_{\ln,\epsilon} = C \ln \frac{x^2 + \epsilon^2}{\epsilon^2} \text{ with } \epsilon = 0.1, \text{ compared to } \|x\|_1 \text{ and thresholded } \|x\|_0.$$

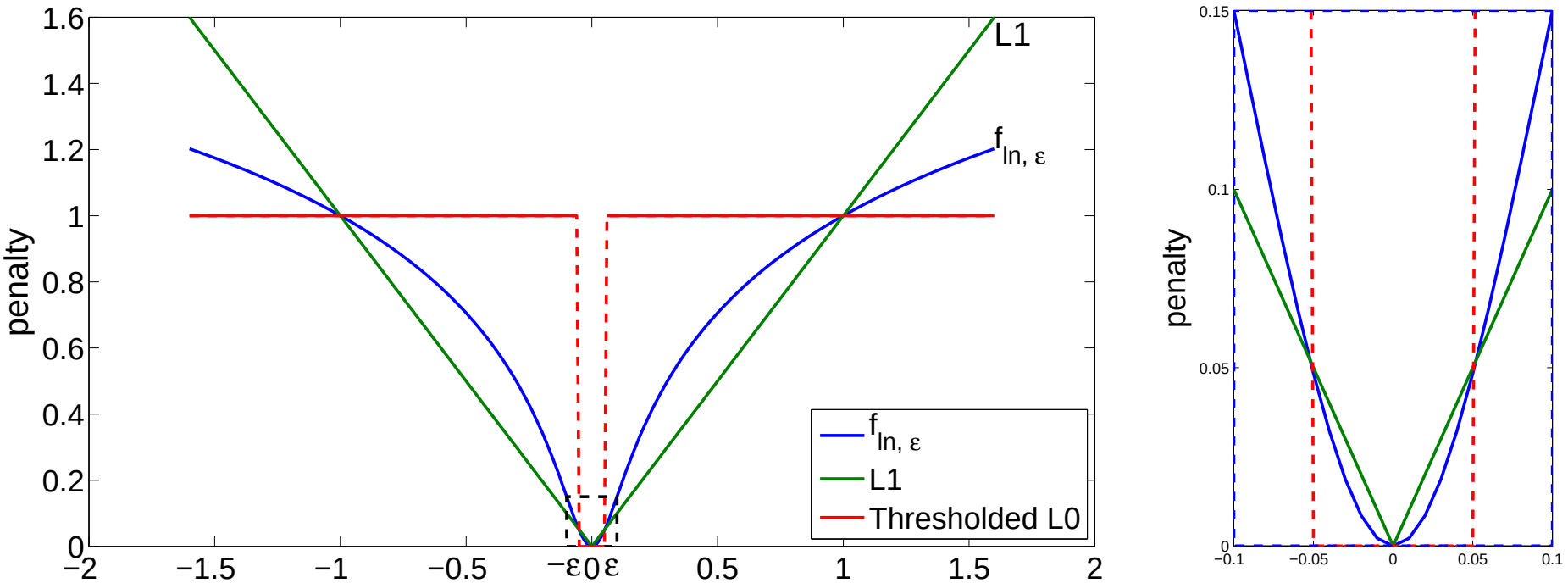


Figure: Compared to the L1 norm, the geometry of the log-sum penalty function lends itself well to detecting sparsity as $\epsilon \to 0$.

Geometry: unit ball of $f_{\ln,\epsilon}(x)$

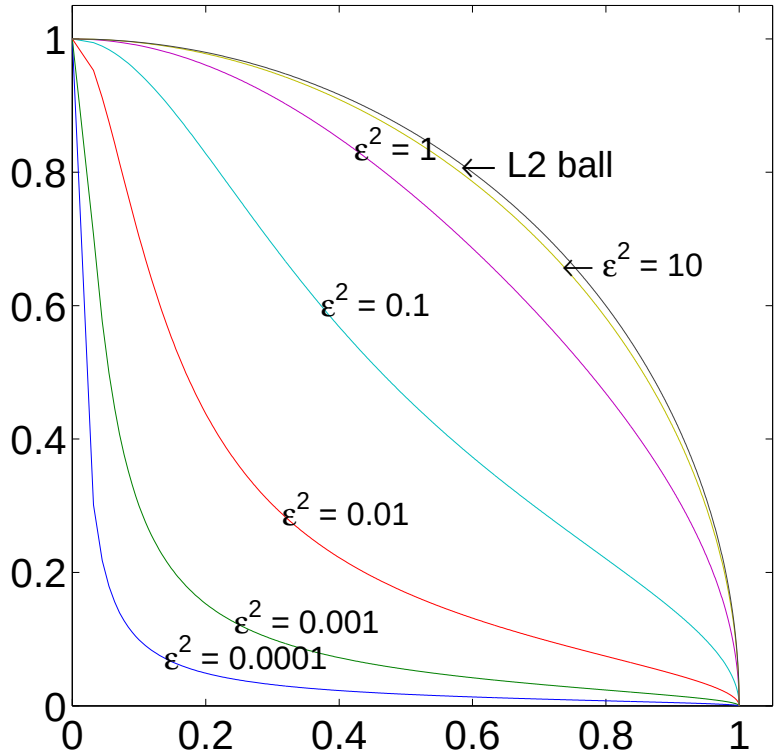


Figure: The effect of ϵ on the geometry of the unit ball of $f_{\ln,\epsilon}(x)$. When ϵ is large, the geometry of $f_{\ln,\epsilon}$ approximates the L2 ball; when ϵ becomes small, the geometry of $f_{\ln,\epsilon}$ approaches that of the L0 norm.

Acceleration and parameter selection

Now we consider the minimization of our problem:

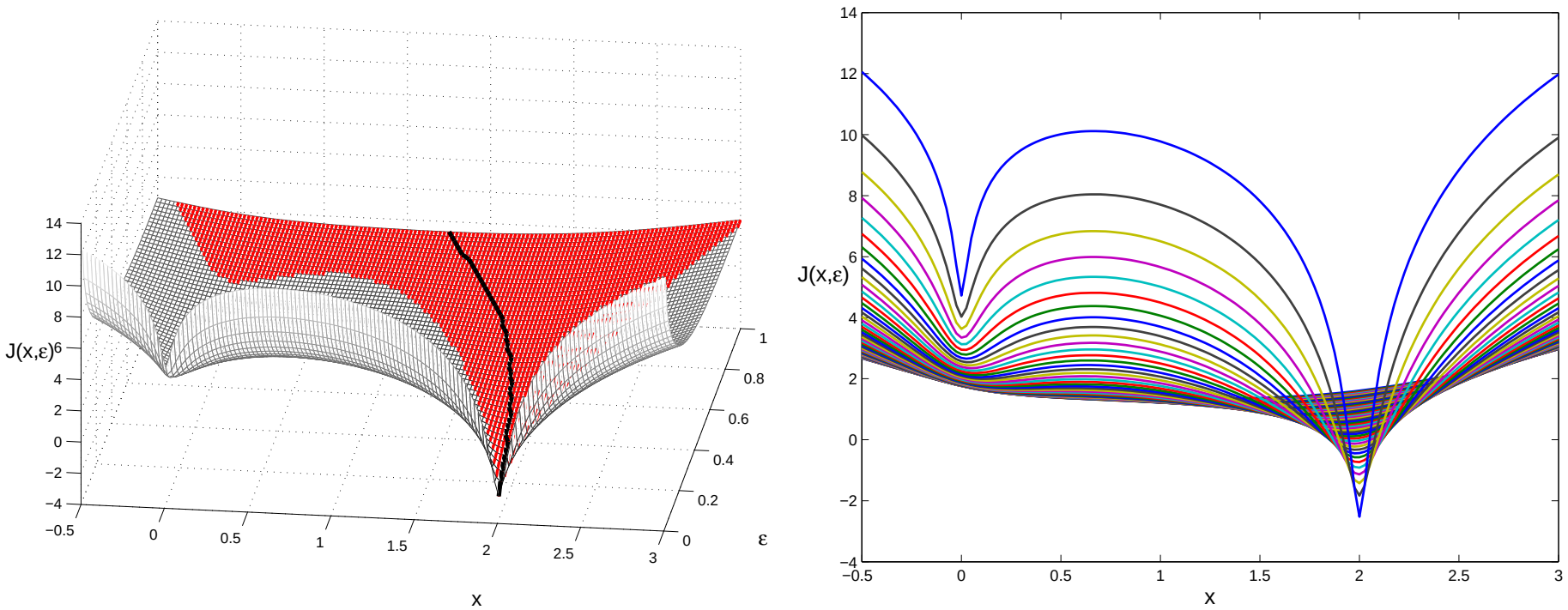
$$F_{\epsilon,\nu}(\mathbf{x}) = \nu^2 \sum_j \ln \frac{|x_j|^2 + \epsilon^2}{\epsilon} + \|\mathbf{y} - \Phi\mathbf{x}\|^2, \tag{10}$$

where there are two parameters which decide the solution path. ν balances the fidelity and sparsity, and ϵ decides the geometry of the penalty function. We formulate a sequence of such problems $F_{\epsilon_n,\nu_n}(\mathbf{x})$:

$$F_{\epsilon_n,\nu_n}(\mathbf{x}) = \nu_n^2 \sum_j \ln \frac{|x_j|^2 + \epsilon_n^2}{\epsilon_n} + \|\mathbf{y} - \Phi\mathbf{x}\|^2 \tag{11}$$

starting from large ν_0 and ϵ_0, then simultaneously reducing ν_n and ϵ_n until $\nu_n = \nu$ and $\epsilon_n = \epsilon$. It is easy to see that $F_{\epsilon_n,\nu_n}(\mathbf{x})$ continuously deforms to $F_{\epsilon,\nu}(\mathbf{x})$. We try to ensure that the path of the global minima of $F_{\epsilon_n,\nu_n}(\mathbf{x})$ leads to the global minimum of $F_{\epsilon,\nu}(\mathbf{x})$.

Geometry: changes of the multidimensional cost function



(a) $J(\mathbf{x}, \epsilon)$. The region where $\frac{\partial J(\mathbf{x},\epsilon)}{\partial \epsilon} > 0$ is coloured red.

(b) Projection of $J(\mathbf{x}, \epsilon)$ to plane $\epsilon = 0$

Figure: The geometry changes of $J(\mathbf{x}, \epsilon) = \sum_{j=1}^{3} \sum_j \ln(x_j^2 + \epsilon^2)/\epsilon$ on a toy example: $\Phi = [3\ 2\ 3\ ;3\ 3\ 1.5]$ and $\mathbf{y} = [6; 6]$. The bold line connects all global minima as ϵ reduces towards zero. Sparsest solution is $\mathbf{x} = [2; 0; 0]$.

Algorithm with continuation

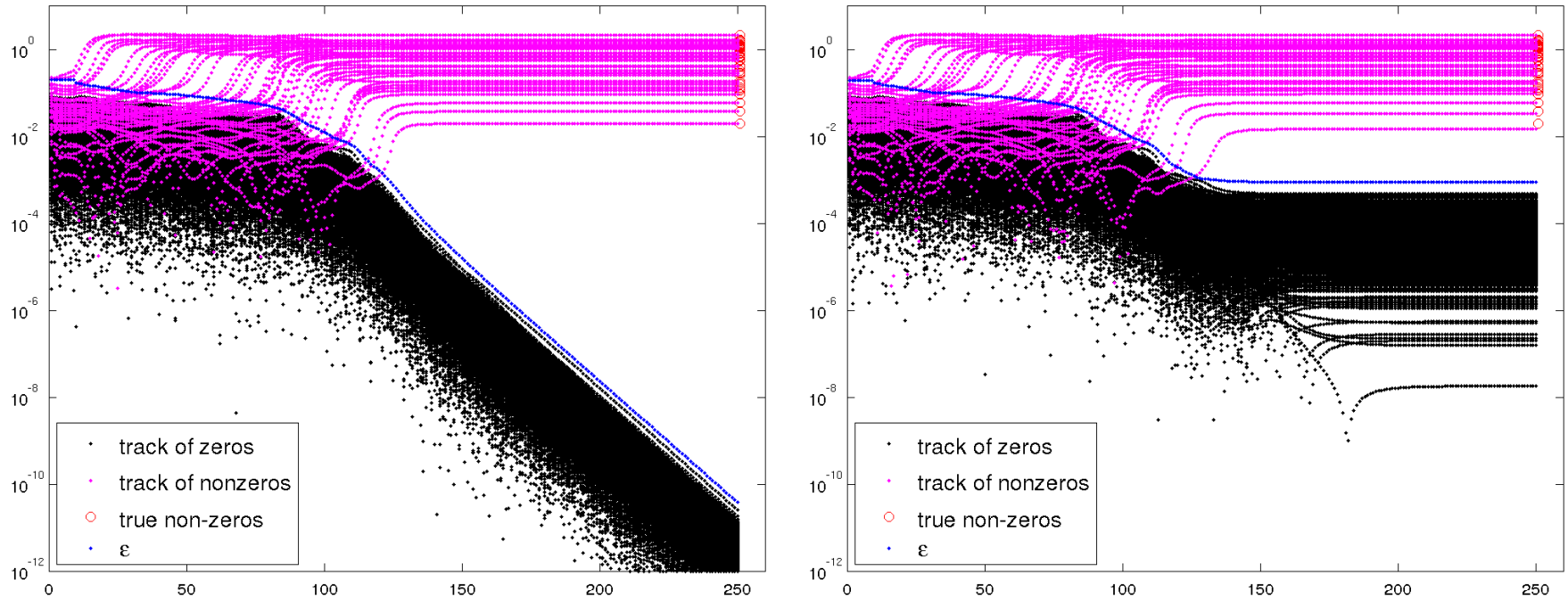
Let $r_L(\mathbf{x})$ denote the Lth largest amplitude element of vector $\mathbf{x}$ and $L_{\max}$ be the maximum number of nonzero terms we expect in $\mathbf{x}$.

Set the initial $\epsilon = \|\Phi^T \mathbf{y}\|_\infty$, and then reduce ϵ gradually.

1. Estimate $\mathbf{x}_{n+1}$ and $\mathbf{z}_{n+1}$ using eq(7a)
2. Update ϵ and ν:
   $L = \min(L + \Delta L, L_{\max})$
   $\epsilon = \min(2r_L(\mathbf{x}_{n+1}), \epsilon)$
   $\nu^2 = 8\epsilon^2 \min(\boldsymbol{\alpha})$ to ensure local convexity of $F_{\epsilon,\nu}(\mathbf{x})$
3. Update $\mathbf{S}$ according to eq(7b)

ΔL controls the convergence rate and is chosen according to the size of the worst-case sparsity $L_{\max}$ and the desired tradeoff between convergence rate and probability of reaching the global minimum.

1-D random signal



(a) noise free

(b) noise s.d. $= 0.05$, SNR ≈ 42dB

Figure: Plots showing how $\mathrm{L_0RL_2}$ gradually selects the non-zero components of $\mathbf{x}$ as ϵ is reduced. The sparse input signal has 1500 elements, among which there are 45 non-zeros. This signal is similar to that used by Daubechies et al in [2]. In the example (b) with noisy observations, ν converges to 0.0576, whereas the true s.dev of the added noise is 0.05.

1-D random signal: noise free example

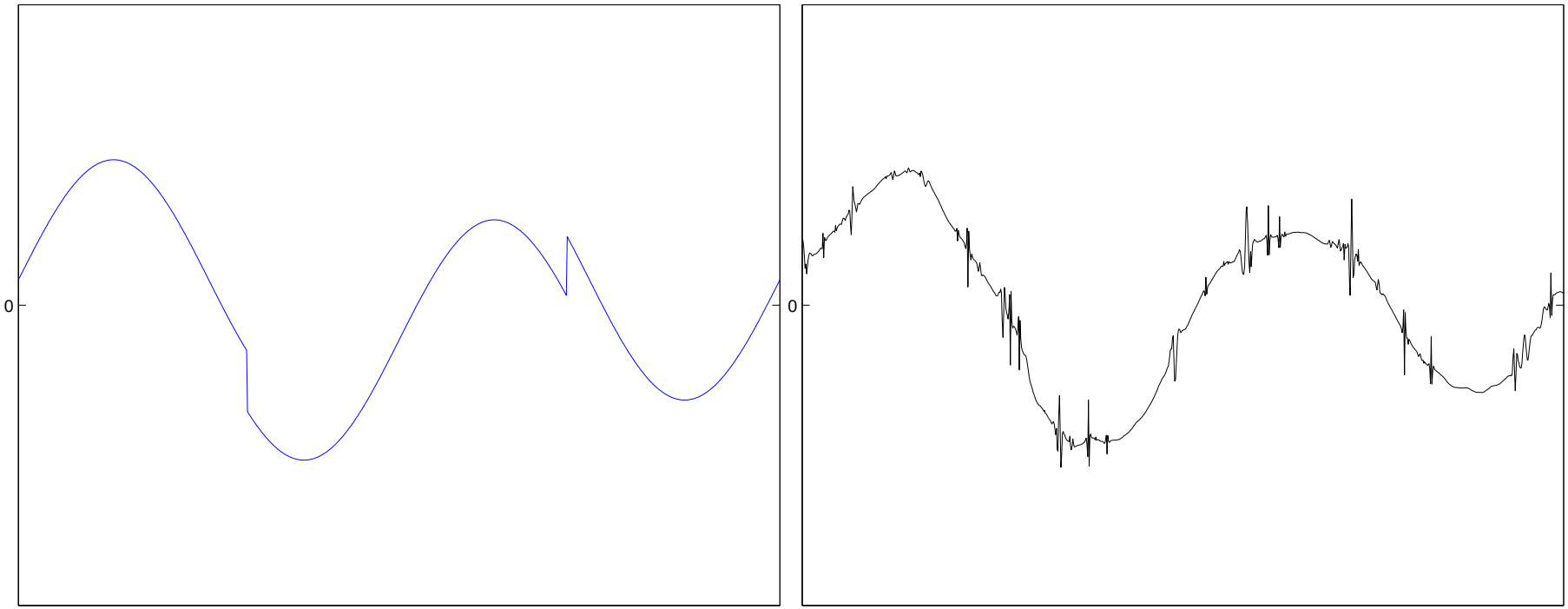
Table: Time comparison of different algorithms

	RMSE	Time (s)	RMSE	Time (s)
L0RL2	8.80E-3	**0.11**	**8.80E-7**	**0.19**
IRLS			9.60E-5	32.50
SL0	5.20E-3	0.61	2.29E-5	0.84
IRL1			2.80E-6	0.59
L1-SpaRSA	8.90E-3	0.18	2.50E-4	0.67
L1-SPGL1	9.10E-3	0.31	1.20E-5	1.30

$$\mathrm{RMSE} = \|\mathbf{x} - \mathbf{s}\|_2 / \|\mathbf{s}\|_2, \tag{12}$$

where $\mathbf{x}$ is the final estimation and $\mathbf{s}$ is the true input signal.

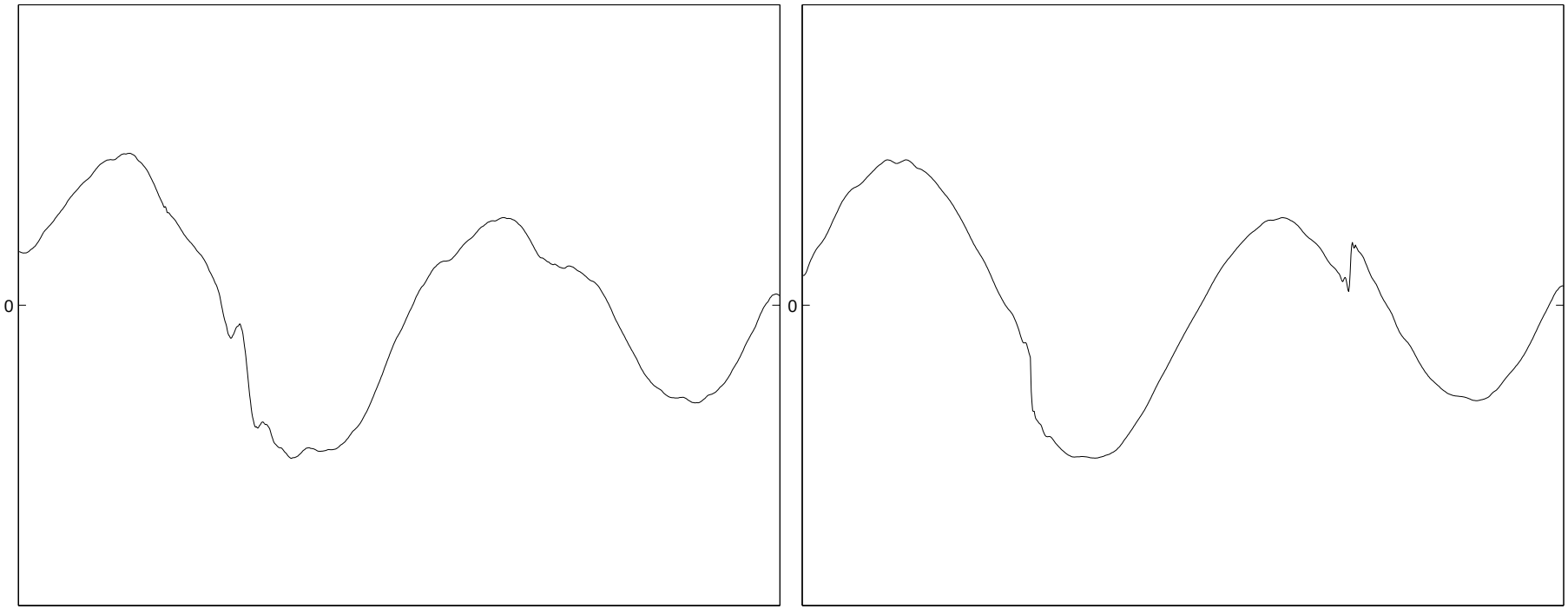
1-D Heavisine signal



(a) Original signal (b) L1-optimization

Figure: The heavisine signal has sparse representation on wavelet basis, which is well structured. Utilizing such information in signal reconstruction often results in improved performance. We use dual-tree complex wavelets (DT-CWT) for optimal performance (see our paper [4] for results with structural constraints).

1-D Heavisine signal



(c) IRL1, $\mathrm{RMSE} = 0.042$ (d) $\mathrm{L_0RL_2}$, $\mathrm{RMSE} = 0.026$

Figure: The heavisine signal has sparse representation on wavelet basis, which is well structured. Utilizing such information in signal reconstruction often results in improved performance. We use dual-tree complex wavelets (DT-CWT) for optimal performance (see our paper [4] for results with structural constraints).

Super-resolution

A real problem:
In some situations the point-spread function (psf) **H** at a higher sampling rate is available, although **y** at the same sampling rate is not, e.g. low inter-slice sampling rate in 3D microscope or medical scan data.
Assume the observation is only available at a lower sampling rate, which we model as:

$$\mathbf{D}(\mathbf{y}) = \mathbf{D}(\mathbf{H}\mathbf{x}) + \mathbf{D}(\mathbf{n}) \tag{13}$$

where **D** is a matrix that represents the subsampling operation.
For simplicity, we denote this as:

our model

$$\bar{\mathbf{y}} = \mathbf{D}(\mathbf{H}\mathbf{x}) + \bar{\mathbf{n}} \tag{14}$$

Embedding tree structure into the algorithm

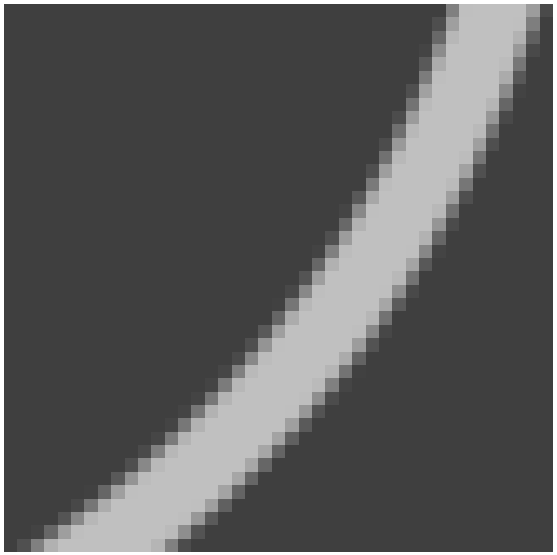
To further enhance the algorithm we can fairly easily build structural tree dependencies into the reweighting matrix $\mathbf{S}$, which serves as a regularization constraint to confine the support of the large coefficients. This is especially effective with the DT-CWT because of its shift-invariance, its complex coefs, and its directionally selective filter properties.

Tree-model options

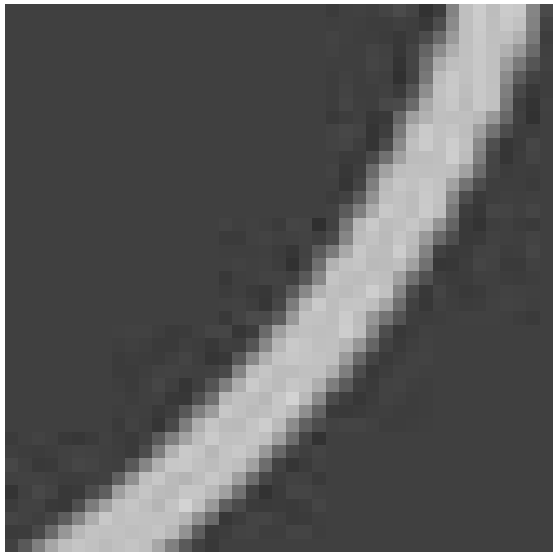
- Enforce the tree structure in the model to generate $\mathbf{S}$
- Use bivariate shrinkage to denoise $\mathbf{W}\mathbf{x}^{(n+1)}$, and then use the denoised coefficient energies to update $\mathbf{S}$.

We find the second method gives the best results so far ...

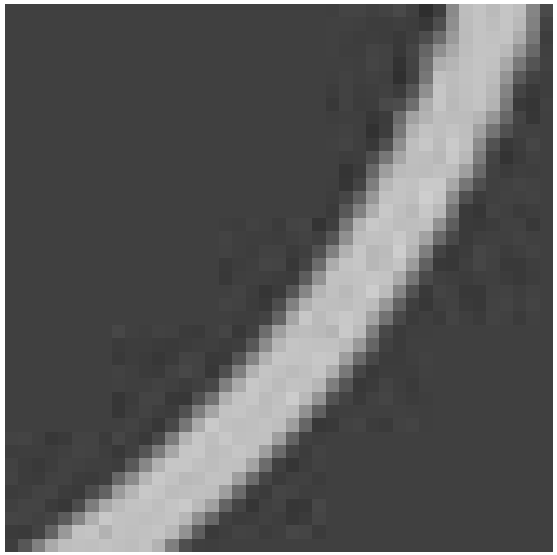
Super-resolving a simple ring in 2D:



(part of) original

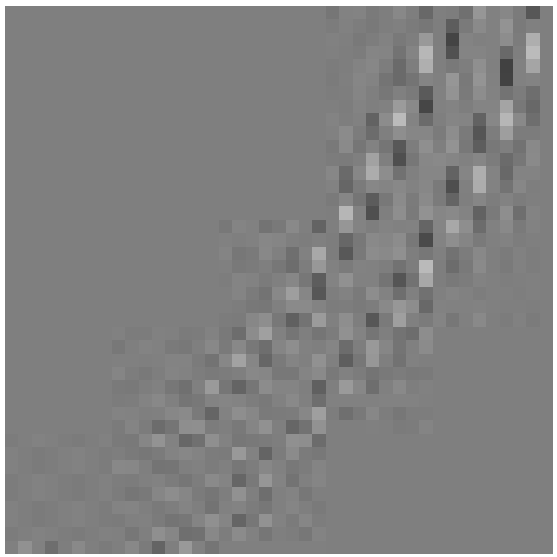


cubic spline

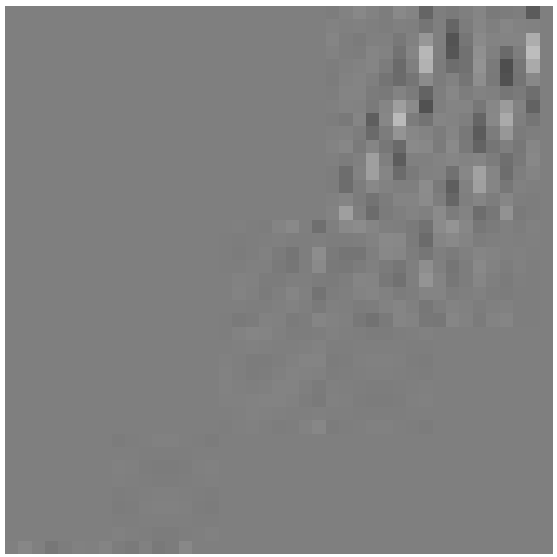


$\mathrm{L_0RL_2}$

Alternate columns are omitted from the original



Difference from original:



Results on Cameramen

Gaussian filter, $[16 \times 16]$, $\sigma_f = 1$ pel, BSNR $= 40$dB.

(a) Observation with even columns missing

(b) odd columns of (a)

Results on Cameramen

Gaussian filter, $[16 \times 16]$, $\sigma_f = 1$ pel, BSNR $= 40$dB.

(c) Cubic Spline interpolation of (d)

(d) deconvolved result from (b)

Results on Cameramen

Gaussian filter, $[16 \times 16]$, $\sigma_f = 1$ pel, BSNR $= 40$dB.

(e) restored to higher resolution

(f) odd columns of (e)

Results on Cameramen: quantitative results

We use ISNR to quantify the improvement:

$$\mathrm{ISNR}(\mathbf{x}^{(n)}) = 10\log_{10}\left(\frac{\|\mathbf{y}-\mathbf{x}\|^2}{\|\mathbf{x}^{(n)}-\mathbf{x}\|^2}\right)$$

$$\mathrm{ISNR}(\mathbf{D}(\mathbf{x}^{(n)})) = 10\log_{10}\left(\frac{\|\bar{\mathbf{y}}-\mathbf{D}\mathbf{x}\|^2}{\|\mathbf{D}\mathbf{x}^{(n)}-\mathbf{D}\mathbf{x}\|^2}\right)$$

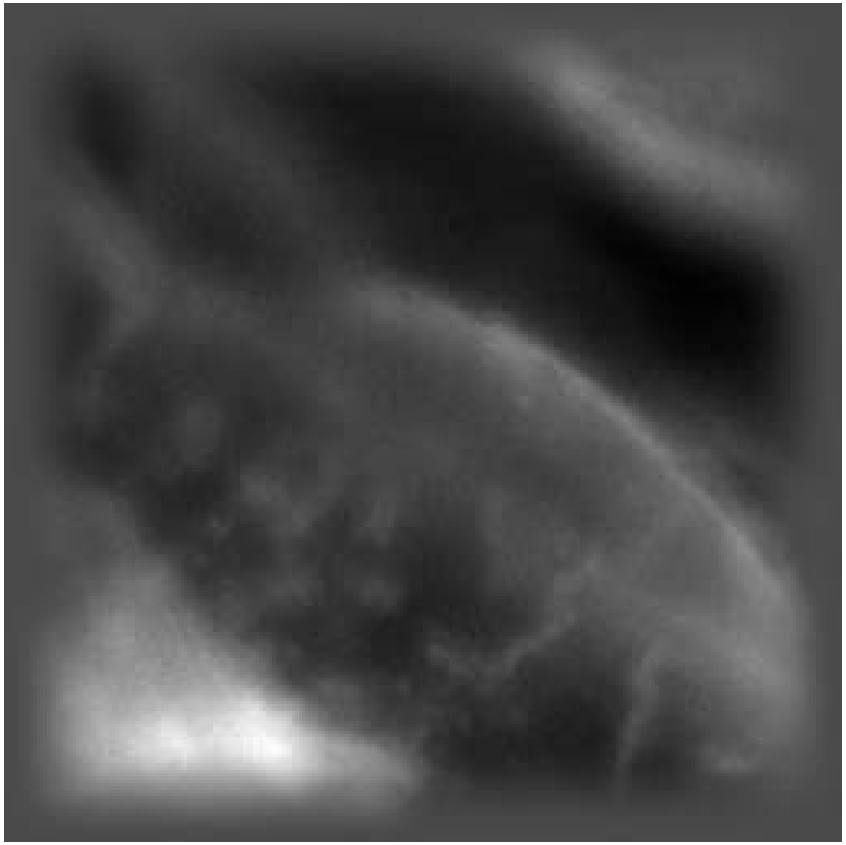
and get the following ISNR(dB) measures:

	(c)	(d)	**(e)**	**(f)**
Gaussian filter 16×16	3.1734	3.2720	**4.2188**	**4.0900**
Uniform blur filter 8×8	4.3641	4.5664	**5.5276**	**5.5179**

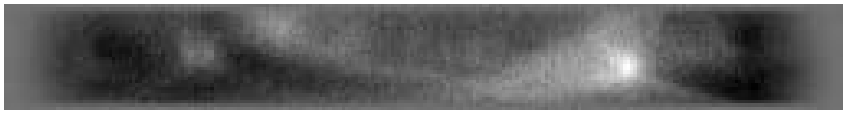
3D microscopy dataset

- Wide-field fluorescence 3D datasets are often not equally sampled in all directions to save time and cost. (Similarly for many medical 3D datasets.)
- The 3D DT $\mathbb{C}$WT representation of the dataset gives 28 planar-wave oriented subbands in each scale. These emphasize planar and linear object features, while noise and aliasing effects in other directions are suppressed by the sparsity-inducing regularization process.
- The coefficients in different subbands are denoised by bivariate shrinkage [Sendur & Selesnick 2002] to improve the quality of $\mathbf{S}$, and thus the final result.

Original image

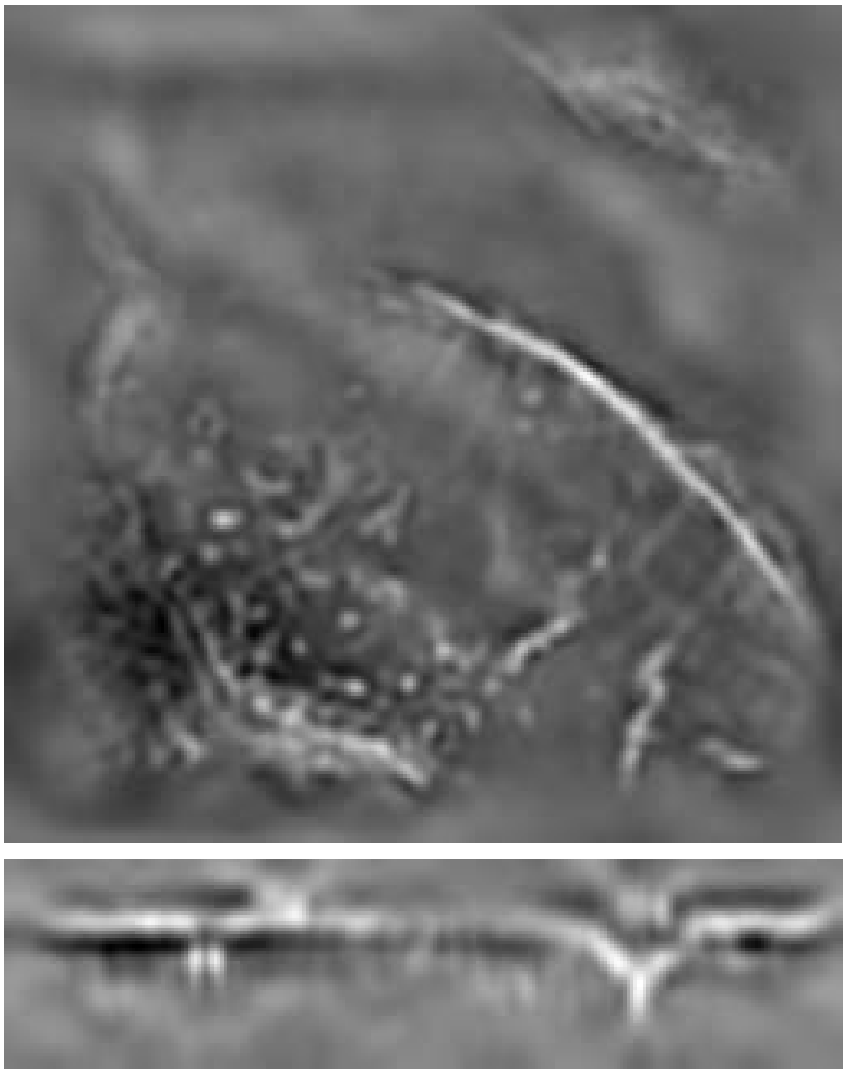


horizontal

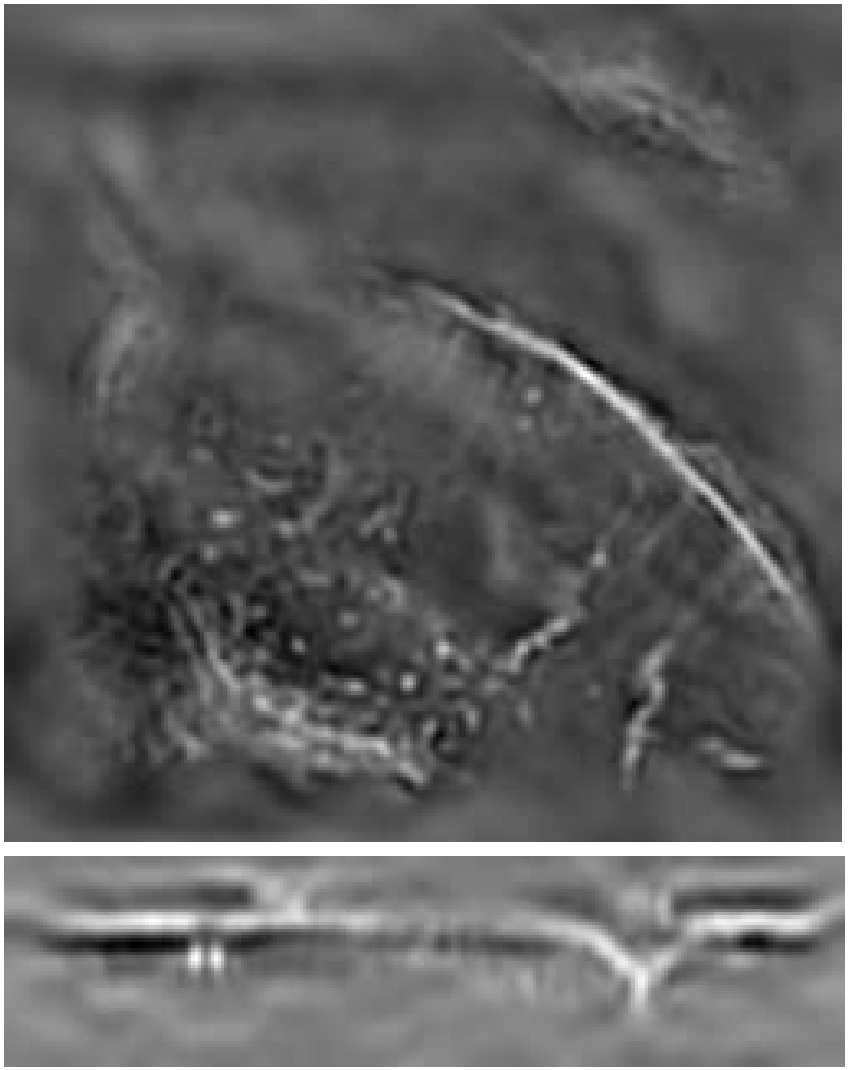


vertical

Spline interpolated image



Our recovery - more results available on our website



Conclusion

The $\mathrm{L_0RL_2}$ algorithm

- is suitable for large-scale problems, e.g. 3D datasets, because it only requires matrix-vector multiplications and element-wise operations in each iteration. Convergence is fast for many problems.
- relies (in its basic form) only on one preset parameter, the sparsity level $L_{\max}$. A loose estimate of $L_{\max}$ is enough to achieve good results.
- is easy to implement and very flexible to allow the integration of prior knowledge of signal structure (see [4] for explanation and results).
- is well-suited to handling noisy measurement data (because of its L_2 heritage), and provides good noise suppression in the final results.

Our model turns out to be similar to hierarchical (sparse) Bayesian modelling, often solved by the Relevance Vector Machine [1].

References

C.M. Bishop and M.E. Tipping.

Variational relevance vector machines.

In *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, pages 46–53. Citeseer, 2000.

I. Daubechies, R. DeVore, M. Fornasier, and S. Gunturk.

Iteratively re-weighted least squares minimization for sparse recovery.

Communications on Pure and Applied Mathematics, 2009.

Cédric Vonesch and Michael Unser.

A fast thresholded landweber algorithm for wavelet-regularized multidimensional deconvolution.

IEEE Transactions on Image Processing, 17(4):539–549, 2008.

Y. Zhang and N. Kingsbury.

Fast l0-based sparse signal recovery.

In *IEEE International Workshop on Machine Learning for Signal Processing (MLSP 2010)*,, Kittila, Finland., Aug 29 - Sept 1 2010.