Optimally Tuned Iterative Reconstruction Algorithms for Compressed Sensing

Arian Maleki
Department of Electrical Engineering
Stanford University
arianm@stanford.edu

David L. Donoho
Department of Statistics
Stanford University
donoho@stanford.edu

Abstract— **We conducted an extensive computational experiment, lasting multiple CPU-years, to optimally select parameters for two important classes of algorithms for finding sparse solutions of underdetermined systems of linear equations. We make the optimally tuned implementations freely available at `sparselab.stanford.edu`; they run 'out of the box' with no user tuning: it is not necessary to select thresholds or know the likely degree of sparsity.**

Our class of algorithms includes iterative hard and soft thresholding with or without relaxation, as well as CoSaMP, Subspace Pursuit and some natural extensions. As a result, our optimally tuned algorithms dominate such proposals.

Our notion of optimality is defined in terms of phase transitions, i.e. we maximize the number of nonzeros at which the algorithm can successfully operate. We show that the phase transition is a well-defined quantity with our suite of random underdetermined linear systems. Our tuning gives the highest transition possible within each class of algorithms. We verify by extensive computation the robustness of our recommendations to the amplitude distribution of the nonzero coefficients as well as the matrix ensemble defining the underdetermined system.

Several specific findings are established. (a) For all algorithms the worst amplitude distribution for nonzeros is generally the constant-amplitude random-sign distribution; where all nonzeros are the same size. (b) Various random matrix ensembles give the same phase transitions; random partial isometries give different transitions and require different tuning; (c) Optimally tuned Subspace Pursuit dominates optimally tuned CoSaMP, particularly so when the system is almost square. (d) For randomly decimated 2D partial Fourier transform sampling, optimally-tuned Iterative Soft Thresholding gives extremely good performance.

I. INTRODUCTION

A recent flood of publications offers numerous schemes for obtaining sparse solutions of underdetermined systems of linear equations; a long list of useful ideas and suggestions can be gleaned from the papers [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [21], [17], [19], [20], [22], [23], [24], [25], [26], with new proposals appearing regularly. Such algorithms have potential applications in fields ranging from medical imaging to wireless digital communications.

The interested user now has a bewildering variety of ideas and suggestions that might be helpful, but this, paradoxically, creates uncertainty and may cause a potential user of such algorithms to avoid the topic entirely. In this note we announce a solution to this problem, in the form of freely available, optimally tuned algorithms ready for use 'out of the box'. Our tuning is based on a comprehensive study of parameter variations and options. It would have required several years to complete our study on a single modern desktop computer. Optimal tuning manages to make some very simple and unsexy ideas perform surprisingly well, reducing the need for more ambitious and impressive sounding ones (even if optimally tuned). Our tuning is based on quantitative principles, it can be used for other algorithms as well and implicitly establishes the 'current state of the art' which future proposals may be compared against. It also generates insights previously unavailable about performance comparisons of methods and performance comparisons of different matrix ensembles.

The empirical tuning approach has a larger significance for the field of sparse representations and compressed sensing. Many of the better known papers in this field discuss *what can be proved rigorously*, using mathematical analysis. It requires real mathematical maturity to understand what is being claimed and what the interpretation must be, and to compare claims in competing papers. Often, what can be proved is vague (with unspecified constants) or very weak (unrealistically strong conditions are assumed, far from what can be met in applications). For practical engineering applications it is important to know *what really happens* rather than what can be proved. Empirical studies of phase transitions provide a direct method to give engineers useful guidelines about what really does happen.

The empirical tuning approach also addresses a difficulty many potential users face in addressing the large and growing literature in sparse representations and compressed sensing. In that literature, a rich variety of brand names is developing, where small variations in some already well-known algorithmic schema may lead to the introduction of extravagant new acronyms and phrases. The outsider will be wary of investing the time to digest all this literature and form an accurate understanding of the differences. Empirical tuning efforts group together several differently-named ideas within one family of algorithmic schemas and optimize settings across the whole family, thereby simplifying the situation for many potential users, since the recommended algorithm both runs 'out of the box' and has been tuned to supersede several different earlier proposals.

II. ITERATIVE ALGORITHMS

Our problem setting will be described with the following notation. An unknown vector $x_0 \in \mathbb{R}^N$ is of interest; we have measurements $y = Ax_0$. Here A is an $n \times N$ matrix and $N > n$. Although the system is underdetermined, it has been shown that sufficient *sparsity* of x_0 may allow unique solution. We say that x_0 is k-sparse if it has at most k nonzeros. In many cases one can exactly recover such a sparse solution x_0 as the solution to

$$(P_1) \min \|x\|_1 \text{ subject to } y = Ax,$$

where $\|x\|_1$ denotes the ℓ_1 norm. This amounts to a large-scale linear programming problem. Unfortunately in some interesting potential applications [27], [28], the matrix A and vector x_0 may contain millions of entries and standard linear programming codes may be too slow in those applications. Hence there is widespread interest in finding fast algorithms that work essentially as well; in particular applications work by Jean-Luc Starck and co-authors [6], [8], [18] and by Michael Elad and co-authors [9], [20] has shown that some very simple iterative algorithms can be strikingly successful on very large problems [29], [20]. In this note we consider two families of such iterative algorithms.

A. Simple Iterative Algorithms

The first family is inspired by the classical relaxation method for approximate solution of large linear systems. In classical relaxation, one iteratively applies A and A' to appropriate vectors and under appropriate conditions, the correct solution is obtained as a limit of the process. While the classical theory is inapplicable to underdetermined systems, it has been found both empirically and in theory that a sparsity-promoting variant of relaxation can correctly solve such systems, when they have sparse solutions. Starting from $x_1 = 0$, one repeatedly applies this rule:

$$x_{i+1} = \eta(x_i + \kappa \cdot (A'r_i)); \qquad r_i = y - Ax_i;$$

Here κ is a relaxation parameter $(0 < \kappa < 1)$ and η is a nonlinear thresholding rule; we consider both Hard thresholding – $\eta_t^H(y) = y1_{\{|y|>t\}}$ and Soft thresholding $\eta_t^S(y) = sgn(y)(|y| - t)_+$. Note that if we set $\eta(y) = y$ we would just have classical relaxation. Iterative Soft Thresholding with a fixed threshold has been used in various settings more than a decade ago – see for example published work of Sylvain Sardy and co-authors [3]. A formal convergence analysis was given by [5] in the determined case. Iterative Hard Thresholding (IHT) was reported useful for several underdetermined cases by Starck, Elad, and their co-auuthors in papers appearing as early as 2004, [6], [8], [9], [18], [20] often outperforming IST. Other recent examples of such ideas include [13], [40], [21], [22].

These iterative schemes are easy to implement: they require only two matrix-vector products per iteration and some vector additions and subtractions. For certain very large matrices we can rapidly apply A and A' without representing A as a full matrix – examples include partial Fourier and Hadamard transforms. In such settings, the work required scales very favorably with N eg as (eg $N\log(N)$ flops rather than $O(N^2)$).

Actually using such a scheme in practice requires choosing a parameter vector $\theta = (type, \kappa, t)$ here type = S or H depending as soft or hard thresholding is required; the other parameters are as earlier. Moreover t needs to vary from iteration to iteration. The general form in which such schemes are often discussed does not give a true ready-to-run algorithm. This is akin to presenting a recipe listing ingredients for a dish, without listing the needed amounts; it keeps potential users from successfully exploiting the idea.

B. Composite Iterative Algorithms

In solving determined linear systems, relaxation can often be outperformed by other methods. Because of the similarity of relaxation to IST/IHT schemes, parallel improvements seem worth pursuing in the sparsity setting. A more sophisticated scheme – Two Stage Thresholding – uses exact solution of small linear systems combined with thresholding before and after this solution. In stage one, we screen for 'significant' nonzeros just as in IST and IHT:

$$v_i = \eta^1(x_i + \kappa A'r_i); \qquad r_i = Y - Ax_i;$$

We let I_i denote the combined support of v_i and x_i and we solve

$$w_i = (A'_{I_i}A_{I_i})^{-1}A'_{I_i}y.$$

We then threshold a second time,

$$x_{i+1} = \eta^2(w_i),$$

producing a sparse vector. Here the threshold might be chosen differently in stages 1 and 2 and might depend on the iteration and on measured signal properties.

It seems that the use of explicit solutions to the smaller systems might yield improved performance, although at the cost of potentially much more expense per iteration. An important problem is user reticence. In the case of TST there are even more choices to be made than in the case to be made with IST/IHT. This scheme again presents the 'recipe ingredients without recipe amounts' obstacle: users may be turned off by the requirement to specify many such tunable parameters.

C. Threshold Choice

Effective choice of thresholds is a heavily-developed topic in statistical signal processing. We have focused on two families of tunable alternatives.

Interference Heuristic. We pretend that the marginal histogram of $A'r$ at sites in the coefficient vector where $x_0(i) = 0$ is Gaussian, with common standard deviation. We estimate the standard deviation of $A'r$ robustly at a given iteration and set the threshold t as a fixed multiple $\lambda \cdot \sigma$. The underlying rationale for this approach is explained in [14] where its correctness was heavily tested. Under this heuristic, we control the threshold λ as in standard detection theory using the False Alarm Rate; thus $2 \cdot FAR = \Phi(\lambda)$ where Φ denotes the standard normal distribution function.

Oracle Heuristic. In the TST scheme imagine that an oracle tells us the true underlying sparsity level k, and we scale the threshold adaptively at each iteration so that at stage 1 we yield $\alpha \cdot k$ nonzeros and at stage two $\beta \cdot k$ nonzeros. The method CoSaMP [23] corresponds to $\beta = 2$, $\alpha = 1$, while Subspace Pursuit [24] corresponds to $\beta = \alpha = 1$.

A problem with the Oracle Heuristic is that, in interesting applications, there is no such oracle, meaning that we wouldn't in practice ever know what k to use. A problem with the Interference Heuristic is that the Gaussian model may not work when the matrix is not really 'random'.

III. PHASE TRANSITIONS

In the case of ℓ_1 minimization with A a random matrix, there is a well-defined 'breakdown point': ℓ_1 can successfully recover the sparsest solution provided k is smaller than a certain definite fraction of n. Let $\delta = n/N$ be a normalized measure of problem indeterminacy and let $\rho = k/n$ be a normalized measure of the sparsity. We get a two-dimensional phase space $(\delta, \rho) \in [0, 1]^2$ describing the difficulty of a problem instance – problems are intrinsically harder as one moves up and to the left. Displays indicating success and failure of ℓ_1 minimization as a function of position in phase space often have an interesting two-phase structure, with phases separated the curve $(\delta, \rho(\delta))$, for a specific function ρ.

Let A be a random matrix with iid Gaussian entries and let $y = Ax_0$ with x_0 k-sparse. In [31], [32] one can find explicit formulas for a function ρ definable with the aid of polytope theory and having the following property. *Fix* $\epsilon > 0$. *The probability that* (P_1) *recovers the sparsest solution to* $y = Ax$ *tends to* 0 *or* 1 *with increasing system size according as* $k = n \cdot (\rho(n/N) \pm \epsilon)$. Informally, all that matters is whether $(n/N, k/n)$ lies above or below the curve $(\delta, \rho(\delta))$. This is the conclusion of a rigorously proven theorem that describes asymptotic properties as $N \to \infty$; it also describes what actually happens at finite problem sizes [34]. The empirically observed fraction of successful recoveries decays from one to zero as the problem sparsity $\rho = k/n$ varies from just below the critical

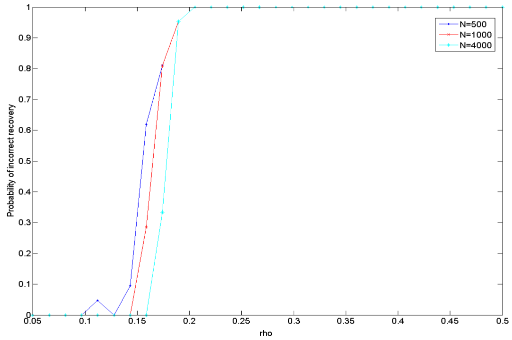

Fig. 1. Fraction of unsuccessful reconstructons by IHT. Here $\delta = .5$ and $\rho = k/n$ is varying. FAR parameter $= 10^{-3}$. Relaxation parameter = 1. At each unique parameter combination, twenty random problem instances were tried. Results are shown at 3 values of N: 500, 1000, 4000

level $\rho(\delta)$ specified in theory to just above it. This transition zone is observed to get increasingly narrow as N increases, matching the theorem, which says that in the large N limit, the zone has vanishing width.

Such sharp phase transitions have also been rigorously proven [30], [33] or empirically observed [14], [16], [36] for other algorithms and/or problem suites. Figure 1 displays behavior of IHT with FAR threshold selection, at a single fixed choice n/N with varying underlying number k of nonzeros. Below a certain threshold, the algorithm works well and above that threshold it fails; the transition zone is narrow, and gets better defined at large problem sizes N.

Incidentally, some readers may be unfamiliar with the notion of phase transitions because popular theoretical tools such as coherence and restricted isometry property do not really give information about them. It has been shown by Jared Tanner and co-authors [35] that bounds derived from RIP ensure the existence of a region with high success probability in the δ-ρ phase space, however, the actual region is much larger than what those bounds provide.

IV. TUNING PROCEDURE

We conducted extensive computational experiments to evaluate the phase transitions of various algorithms. In all, we performed more than $90,000,000$ reconstructions, using 38 servers at a commercial dedicated server facility for one month. These calculations would have run more than 3 years on a single desktop computer.

For a fixed iterative scheme and a fixed tuning parameter θ, we considered in turn each of several problem suites $\mathcal{S} = (E, C)$, i.e. several random matrix ensembles E and several coefficient amplitude distributions C. At each combination, we created numerous problem instances, ran the algorithm, and recorded metrics of algorithm success. We did all this at a wide range of sparsities and indeterminacies.

In the tuning stage of our project we worked only with the *standard suite* $\mathcal{S}_0$ formed with the USE matrix ensemble and constant amplitude distribution on the nonzeros. In the later evaluation stage, other problem suites were used to test the robustness of the tuning.

For a fixed $N = 800$, we varied n and k through a grid of 900 δ and ρ combinations, with δ varying from .05 to 1 in 29 steps and ρ varying from .05 up to a ceiling value $\rho_{max} < 1$ in as many as 30 steps. At each grid point we solved $M = 100$ different random problem instances, obtaining the observed success fraction $Succ(\delta, \rho; \theta, \mathcal{S})$.

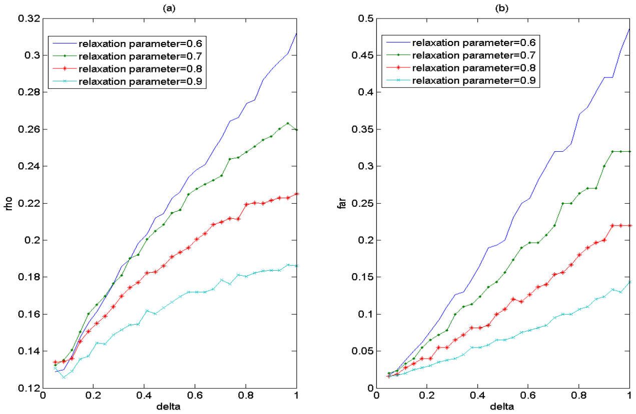

Fig. 2. (a) Optimum phase transitions for IST at several choices of relaxation parameter (b) FAR parameter choice yielding the optimum

We declared a reconstruction successful when the relative ℓ_2 reconstruction error was less than 1 percent. We measured the empirical phase transition by varying k with all other variables held fixed, searching for a value where 50% of the attempted reconstructions were successful. Formally, let $Succ(\delta, \rho; \theta)$ denote the fraction of successes observed for a given algorithm/parameter combination at a specific sparsity-indeterminacy combination; the empirical phase transition is

$$\rho^*(\delta; \theta) = \max\{k/n : Succ(n/N, \rho'; \theta) > .5, \forall \rho' < k/n\}. \quad (1)$$

We denote the optimal parameter choice via

$$\theta^*(\delta) = \arg\max_{\theta} \rho^*(\delta; \theta). \quad (2)$$

V. TUNING RESULTS

Figure 2 illustrates tuning results for IST on the standard suite $\mathcal{S}_0$. Here $\theta = (RelaxationParameter, FARParameter)$. Panel (a) shows the different optimized phase transitions available by tuning FAR to depend on δ while the relaxation parameter is fixed. Panel (b) shows the optimally tuned FAR parameters at each given δ and choice of relaxation parameter. Figures 3 offer the same information for IHT.

Optimum performance of IST occurs at higher values of the false alarm rate than for IHT. Decreasing the relaxation parameter beyond the range shown here does not improve results for IST and IHT.

Figure 5 illustrates tuning results for the variant of TST based on FAR threshold selection. Again $\theta = (RelaxationParameter, FARParameter)$. The phase transitions correspond to different fixed choices of FAR at fixed levels of the relaxation parameter. The results depend very weakly on FAR.

Figure 6 illustrates performance of TST for different values of $\theta = (\alpha, \beta)$. Panel (a) shows the different optimized phase transitions available by tuning β at fixed $\alpha = 1$ and Panel (b) shows optimal phase transitions with $\alpha = \beta$ varying. Both displays point to the conclusion that $\alpha = \beta = 1$ dominates other choices. Hence Subspace Pursuit $(\alpha = 1, \beta = 1)$ dominates CoSaMP $(\alpha = 1, \beta = 2)$.

VI. RECOMMENDED CHOICES

We provide three versions of iterative algorithms based on our optimal tuning exercise: Recommended-IST, Recommended-IHT and

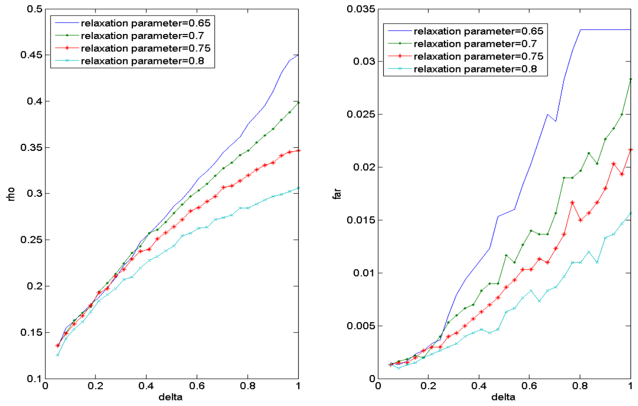

Fig. 3. (a) Optimum phase transitions for IHT at fixed relaxation parameter (b) FAR parameter choice yielding the optimum

Recommended-TST. They are implemented in Matlab and published at URL `sparselab.stanford.edu/ReadyToRun`.

In our recommended versions, there are no free parameters. The user specifies only the matrix A and the left-hand side y. In particular the user does not specify the expected sparsity level, which in most applications cannot be considered known.

These recommended algorithms are not the same as previously published algorithms. For example, Recommended TST has parameters $\alpha = 1$ and $\beta = 1$, so it initially seems identical to Subspace Pursuit [24]. However, Subspace Pursuit demands an *oracle* to inform the user of the true underlying sparsity of the vector. Recommended-TST has already embedded in it a value for the assumed sparsity level at each δ (see Table III). If the actual sparsity in x_0 is better than the assumed value, the algorithm still works, but if it is worse, the algorithm won't work even if tuned to that worse sparsity level. The user does not need to know this number – it is baked into the code. In effect, we have de-oracle-ized the Subspace Pursuit method.

We remind the reader that these algorithms dominate other implementations in the same class. Thus, Recommended TST dominates CoSamp; this is particularly evident for $\delta > .5$ (see Figure 6).

A companion set of algorithms – described later – is available for the case where A is not an explicit matrix but instead a linear operator for which Av and $A'w$ can be computed without storing A as a matrix. Some differences in tuning for that case have been found to be valuable.

We record in the following tables a selection of the optimally tuned parameter values.

TABLE I

RECOMMENDED CHOICES OF FAR AND ρ FOR IST

δ	.05	.11	.21	.31	.41	.5	.6	.7	.8	.93
ρ	.124	.13	.16	.18	.2	.22	.23	.25	.27	.29
FAR	.02	.037	.07	.12	.16	.2	.25	.32	.37	.42

TABLE II

RECOMMENDED CHOICES OF FAR AND ρ FOR IHT

δ	.05	.11	.21	.41	.5	.6	.7	.8	.93
ρ	.12	.16	.18	.25	.28	.31	.34	.38	.41
FAR	.0015	.002	.004	.011	.015	.02	.027	.035	.043

TABLE III

RECOMMENDED ρ FOR TST

δ	.05	.11	.21	.31	.41	.5	.6	.7	.8	.93
ρ	.124	.17	.22	.26	.30	.33	.368	.4	.44	.48

Figure 4 compares our Recommended algorithms with each other and with the non-iterative algorithm of ℓ_1 minimization mentioned in Section III. It gives the empirical phase transitions at the Standard Suite. These transitions obey the following ordering:

$$\ell_1 > \text{Rec-TST} > \text{Rec-IHT} > \text{Rec-IST};$$

this is exactly the ordering which one would expect based on qualitative grounds; however, it is striking to see how close the curves actually are. Both Simple Iterative Algorithms are dramatically less complex to implement and also dramatically cheaper to run on a per iteration basis. It seems that at moderate sparsity levels one would often be satisfied with IHT or IST; particular so for very large problem sizes.

VII. ROBUSTNESS

A *robust* choice of parameters offers a guaranteed level of performance across all situations. Such a choice can be made by solving the maximin problem

$$\theta^r(\delta) = \arg\max_\theta \min_{\mathcal{S}} \rho^*(\delta; \theta; \mathcal{S}).$$

The maximin is achieved at the *least-favorable* suite. Our tuning results were obtained at the standard suite $\mathcal{S}_0$, with constant amplitude, random-sign coefficients and matrices from USE. We considered other suites by varying the matrix ensemble E, including uniformly-distributed random projections and Bernoulli matrices with random signs. We also considered four coefficient ensembles C: in addition to the constant amplitude ensemble, we considered coefficients from

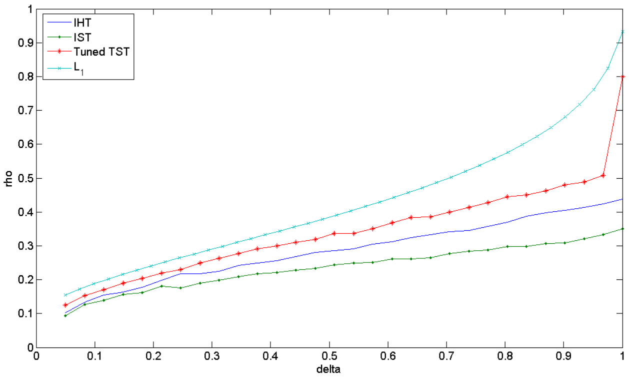

Fig. 4. Phase Transitions of several algorithms at the standard suite. Upper curve: theoretical phase transition, ℓ_1 minimization; lower curves: Observed transitions of algorithms recommended here.

the double exponential distribution, the Cauchy, and the uniform distribution on $[-1, 1]$.

Figures 7-8-9 display results for Recommended-IST, Recommended-IHT, and Recommended-TST. The constant amplitude, random sign ensemble is approximately the least favorable input distribution for all three methods. Since we have tuned at that ensemble, our choice of tuning parameters can be said to be robust.

Figures 10-11-12 study Recommended-IST, Recommended-IHT, and Recommended-TST at three matrix ensembles. Results are similar for the Bernoulli and USE ensembles, and noticeably different – mostly better – for URP. Since we have tuned at USE, our choice of tuning parameters can be called robust.

A surprising exception to the above pattern is described below in Section IX.

VIII. Running Times

Algorithm running times were measured on an "Intel 2 Core Processor 2.13GHz" with 3GBytes RAM. All implementations are in Matlab, and in each case the iterative algorithms were run until RMSE= .01.

TABLE IV

Average running time (sec) until Relative MSE= .01. Standard Suite and $N = 4000$. N^* – problem size at which IHT becomes faster than TST.

δ	.1	.3	.5	.7	.9
IHT	.99	9.31	19.9	34.19	51.4
IST	1.89	13.1	21	36.2	54.3
TST	.07	3.05	30.4	129	412
N^*	57000	12200	2700	1100	500

IX. Ensembles based on Fast Operators

The matrix ensembles discussed so far all used dense matrices with random elements. However, many applications of sparsity-seeking decompositions use linear operators which are never stored as matrices. The partial Fourier ensemble [28] provides an example. Here the $n \times N$ matrix A has for its rows a random subset of the N rows in the standard Fourier transform matrix. Av and $A'w$ can both be computed in order $N \log(N)$ time; the comparable dense matrix vector products would cost order N^2 flops.

The Simple Iterative Algorithms IHT and IST are particularly suited for use in the FastOps setting, since they require only repetitive application of Av and $A'w$ interleaved with thresholding.

We considered three ensembles in the FastOps setting: 1D and 2D partial Fourier ensembles, and 1D partial Hadamard ensemble. We found that

- it is important to tune IHT and IST differently for these ensembles.
- the relaxation parameter in IHT/IST makes essentially no contribution to performance in this setting.
- 1D partial Hadamard and 1D partial Fourier gave very similar results.
- the performance of IHT is very much in line with earlier results for the random matrix ensembles.
- IST behaves dramatically better at partial Fourier ensembles than for the Random matrix ensembles (Figure 14) and even outperforms IHT for $\delta > .5$ (Figure 13).

Recommended parameters are shown in Tables V-VI. Running time of the algorithms for partial Fourier are studied in Table VII

TABLE V

Optimal values ρ and FAR for IST. Partial Fourier ensemble.

δ	.11	.21	.31	.41	.5	.6	.7	.8	.9
ρ	.092	.16	.21	.26	.31	.37	.41	.44	.48
FAR	.0209	.0736	.13	.19	.26	.32	.32	.32	.32

TABLE VI

Optimal ρ and FAR for IHT. Partial Fourier ensemble.

δ	.11	.21	.31	.41	.5	.6	.7	.8
ρ	.14	.2	.24	.27	.3	.32	.34	.38
FAR	.0004	.0018	.0029	.0038	.005	.005	.005	.005

TABLE VII

Running times (sec) for 50 Iterations of Recommended Algorithms. Partial Fourier Ensemble.

N	n	IST, IHT
65536	12000	1.41
262144	24000	6.33
262144	50000	6.53

X. Discussion

Our work makes several contributions.

A. Helping Potential Users

The rapidly growing literature on sparsity-promoting reconstruction methods creates difficulties for potential users who would like to apply that knowledge. For a given problem type, there may be many seemingly relevant papers, each promoting specific techniques that have been labelled with catchy branding. When such papers are based on theoretical analysis, the engineer who is a potential user of such ideas may be overwhelmed by the significant amount of mathematical knowledge required to understand and compare the abstract claims being made in such papers. Even when the papers adopt a more familiar engineering approach, the interest of the paper's authors to emphasize the distinctiveness of their algorithm may lead them to use examples or problem settings that are quite different from those used by other authors, making it quite daunting for a newcomer to compare papers and make decisions about what algorithms might conceivably be useful in the user's setting. One easily imagines that in this situation, some users make attempts to digest some of the literature and get inconsistent or disappointing results, not knowing whether the problem is due to misunderstandings, programming errors, or misapplication of techniques, or true limitations of sparsity-promoting methods.

Our results may help potential users more easily evaluate the potential benefits of well-chosen sparsity-promoting reconstruction methods. Several algorithms are made freely available, their properties are carefully described and compared on a common basis, and the underlying problem suites and performance metrics are available for careful study. We give the user an ability to get started with these ideas very directly and transparently,

B. Defining the State of the Art

Our work may be useful to active researchers in the field of sparsity-promoting methods. Our study effectively defines the *current state of the art* (CSA), a precisely-specified set of quantitative performance standards which are the best we currently know how to do (within a certain class of algorithms). Once this is defined, any newly proposed algorithm can be evaluated with reference to the CSA, and other researchers can use this comparison to understand the relative improvement, if any, offered by the proposal. Over time, as genuine improvements emerge, the CSA will evolve, by definition always offering the best known current performance.

More broadly, a researcher with a new method *not* intended for comparison with the CSA on a standard suite may expand the set of suites beyond those studied here, adding to such a suite a new matrix or coefficient ensemble, or may provide a new measure of success. This allows the researcher to clearly demonstrate for colleagues the arena where the method is intended to contribute.

The effort to create standard performance metrics and define CSA performance has been valuable in many fields of image and signal processing. Indeed, one can argue that in fingerprint recognition and face recognition, the moment when those fields really started to make progress is precisely the moment when defined databases and success metrics were made available for community use, allowing systematic comparison of algorithms [39], [38]. Similarly, the regular publication of standardized challenge problems in arenas like protein structure prediction is said to have utterly transformed the field. Defining a CSA for sparsity-promoting methods can likewise be expected to lead to much faster and more reliable progress.

Standardization has another predictable effect: it may *improve communication among researchers in the field* reducing the impact of marketing, branding and prestige and increasing the focus on objective measures performance. The example of protein structure prediction (CASP) bears this out [37]; before CASP, certain approaches to protein structure prediction were considered more likely to work than others, but it has been reported that CASP upset expectations, reversing the order of preference for certain algorithms. In fact our work may eready be showing this effect: one of our findings is that the more prominent algorithm CoSamp is dominated by the less well-known algorithm SubspacePursuit at the suites we have studied and for the success measure we have used.

C. Promoting Reproducible Computational Research

An implicit but still important contribution of our work is adherence to the paradigm of Reproducible Computational Research [41]. The data and code required to reproduce our results are freely available – and not just the conclusions. A researcher or potential user can study our implementation and tuning of an algorithm or our definitions of performance metric, or our collection of problem suites. This speeds up progress in developing and validating new sparsity-seeking algorithms.

- Researchers who feel our study ignores important aspects of performance to easily define new metrics of success which better reflect their views of what is important, and to easily conduct studies with that new metric.
- Potential users interested in a specific problem suite we haven't studied, but which is of direct interest in their application, may easily extend our software to add a new suite to the available collection and then run a robustness study or even a tuning study focused on that suite.

By sharing the code underlying our study, we promote further developments of new applications and new methods which outperform current ones.

XI. CONCLUSIONS

We defined a set of problem suites and two algorithmic schemes that cover several distinctly branded methods in the literature. We defined algorithm performance using the notion of empirical phase transition and made millions of reconstruction attempts. while systematically varying problem specifications. We identified specific parameter choices which are optimal at so-called standard suites, for specific sparsity-indeterminacy combinations. This produced Recommended-IST, IHT and TST algorithms, coded in Matlab and are freely available at URL `sparselab.stanford.edu/ReadyToRun`. They can be used 'out of the box' on problem instances of the type we have studied; the user need not specify any parameters whatever in order to run them; simply providing the matrix A and the left-hand side y of the system $y = Ax$.

Our studies included extensive computations at other suites besides the standard one, verifying the robustness of our parameter choices and checking that at those other suites the recommended algorithms generally behave better than they do at the standard suite.

The standard suite involves random matrices, but many applications of sparsity-seeking algorithms, particularly in compressed sensing, use structured matrices. We did consider an important class of structured matrices based on fast transforms such as 1D and 2D Fourier transforms and fast Hadamard transforms. Such matrices admit of rapid computations with very large problem sizes and in some cases are actually demanded by the application, for example in MR imaging and NMR spectroscopy. We studied some very large problem sizes with problem suites based on these fast transforms and found results largely matching those we found on the random ensembles. In one case – IST with 2D partial fourier – we found tunings which generate unexpectedly high phase transitions markedly better than what we saw for the standard suite. We published recommended choices of IHT, IST for use with such ensembles defined by those fast operators.

We reached the following empirical findings at the 'random' matrix ensembles

- Phase transitions for optimally-tuned algorithms obey the ordering Recommended-TST $>$ Recommended-IHT $>$ Recommended-IST.
- Setting the relaxation parameter to 0.6 for IST and 0.65 for IHT improves performance of those algorithms significantly. Relaxation has no noticeable effect on performance of TST.
- The performance results for the matrix ensemble USE (start with iid Gaussian entries then normalize column lengths) is very similar to RSE (random ± 1 entries).
- The distribution of coefficient amplitudes in the solution x_0 matters very much to these algorithms. The worst case is when all nonzeros have the same amplitude.
- For a given problem suite, the number of iterations to reach a given reconstruction accuracy does not seem to depend on problem size, except perhaps near to phase transition.
- Subspace pursuit works better than CoSamp on the standard suite. Our recommended TST algorithm is essentially subspace pursuit, but without need for an oracle.

Our conclusions for the case where the matrix is defined using a fast linear operator were listed in Section IX.

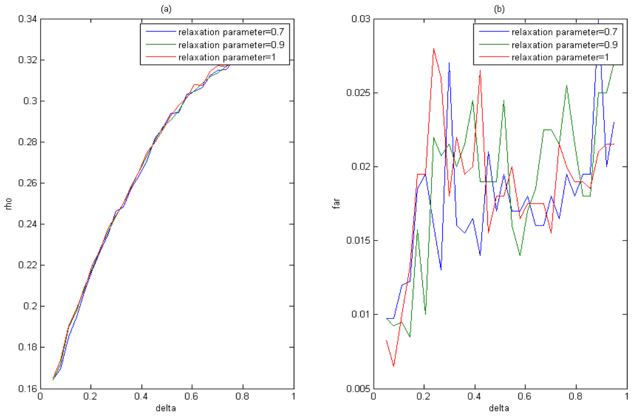

Fig. 5. (a) Optimum phase transitions for TST-FAR at fixed relaxation parameter (b) FAR parameter choice yielding the optimum

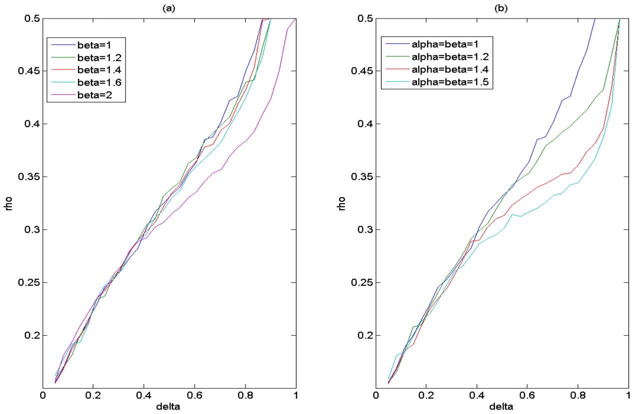

Fig. 6. (a) Empirical phase transitions of TST-(α, β) for $\alpha = 1$ and different values of β; (b) Empirical phase transitions when $\alpha = \beta$.

We were *not* able to discuss the following points:

- Comparisons with standard algorithms OMP, LARS, BP, ℓ_1 minimization and StOMP using the same metrics and ensembles were not available at the moment of submission, but would be interesting to have for comparison purposes.
- Systematic studies at small $\delta = n/N$, $\delta < 0.10$, say, were not attempted on the standard suite; they require very large problem sizes and seem prohibitively slow with dense matrices; they are feasible in the FastOps setting, but were not available at the moment of submission.

ACKNOWLEDGEMENTS

Thanks to Jared Tanner for valuable suggestions on an early draft, and the Special Issue Editors, especially Rick Chartrand, for their service to the IEEE. This work was partially supported by NSF DMS 05-05303.

REFERENCES

[1] I.F. Gorodnitsky, B. D. Rao,"Sparse signal reconstruction from limited data using FOCUSS: a re-weighted minimum norm algorithm," *IEEE*

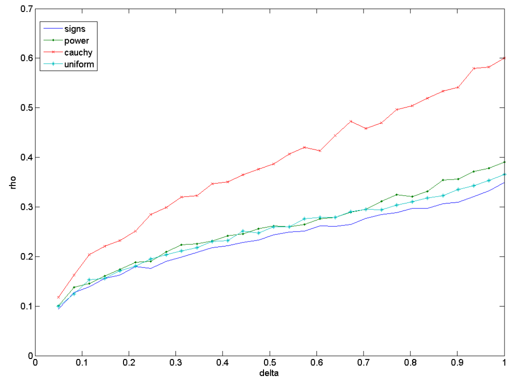

Fig. 7. Observed Phase Transitions of Recommended IST at different coefficient ensembles.

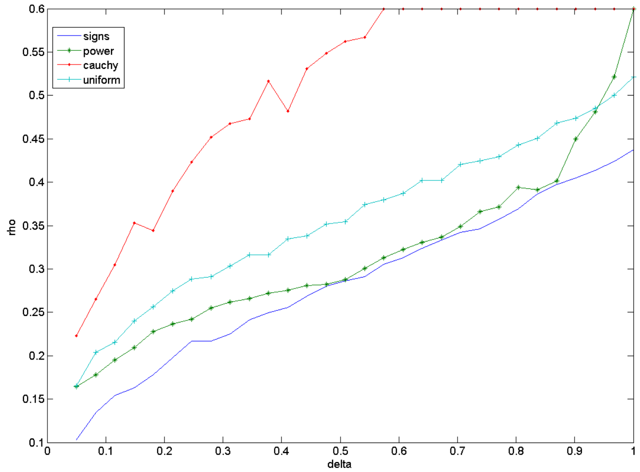

Fig. 8. Observed Phase Transition of Recommended IHT different coefficient ensembles.

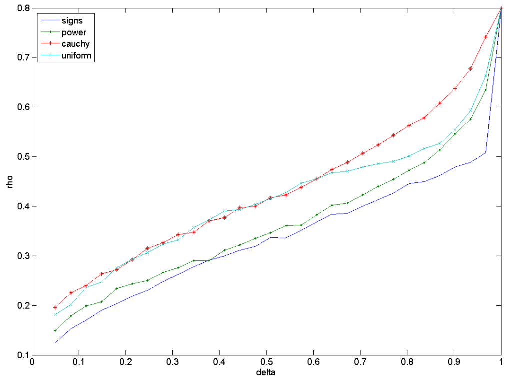

Fig. 9. Observed Phase Transition of Recommended TST for different coefficient ensembles..

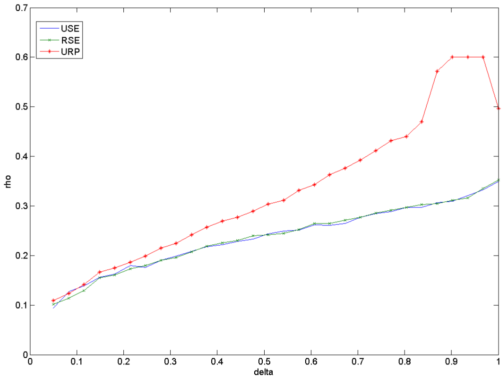

Fig. 10. Observed Phase Transition of Recommended IST for different matrix ensembles.

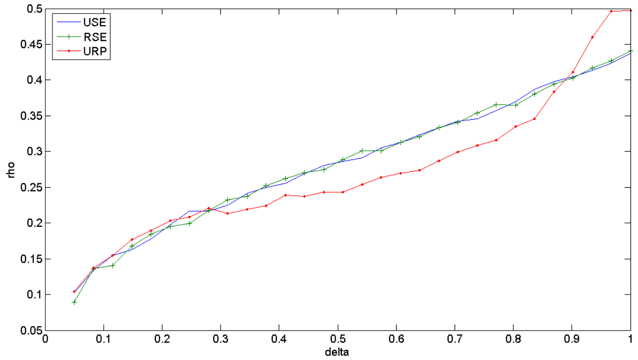

Fig. 11. Observed Phase Transition of Recommended IHT for different matrix ensembles.

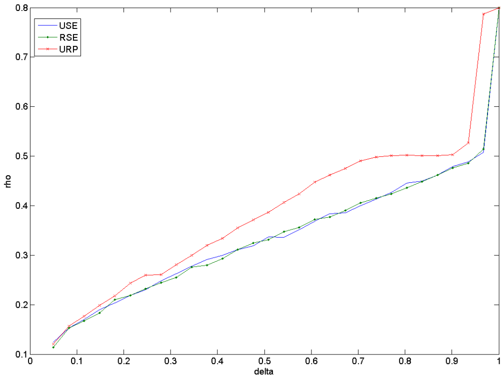

Fig. 12. Observed Phase Transition of Recommended TST for different matrix ensembles.

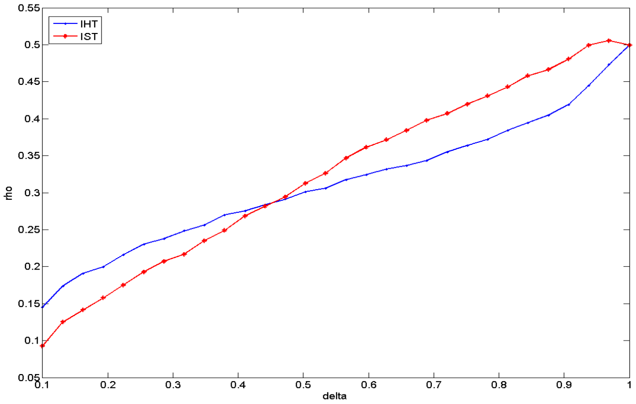

Fig. 13. Comparison of the performance of Recommended IHT and IST for partial fourier ensemble.

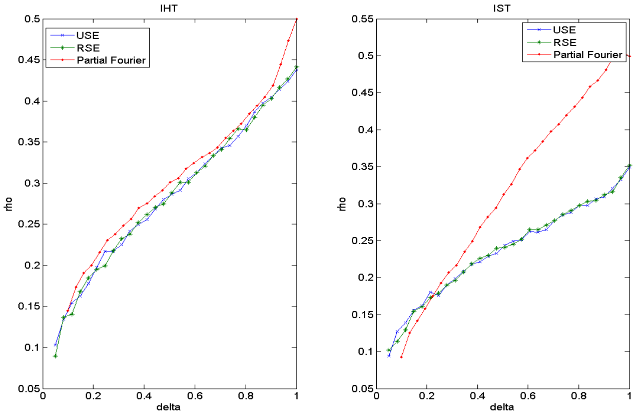

Fig. 14. (a) Phase transitions of recommended IHT for different matrix ensembles (b)Phase transitions of recommended IST for different matrix ensembles

Transactions Signal Processing Volume 45, Issue 3, Mar 1997 Page(s):600 - 616.

[2] S.S. Chen, D.L. Donoho and M.A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, Vol. 20,pp. 33–61,1998.

[3] S. Sardy, A. G. Bruce, P. Tseng, "Block coordinate relaxation methods for nonparametric wavelet denoising," *Journal of Computational and Graphical Statistics*, Vol. 9, pp. 361-379, 2000.

[4] J. Fan, R. Li, "Variable selection via nonconcave penalized likelihood and its oracle properties," *Jour. Ameri. Statist. Assoc.*, 96 (2001), 1348-1360.

[5] I. Daubechies, M. Defrise and C. De Mol "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Communications on Pure and Applied Mathematics*, Vol. 75, pp. 1412-1457, 2004.

[6] J.-L. Starck, M. Elad, and D.L. Donoho, "Redundant multiscale transforms and their application for morphological component analysis", *Journal of Advances in Imaging and Electron Physics*, Vol. 132, pp. 287-348, 2004.

[7] B.,Efron, T. Hastie,I. Johnstone,R. Tibshirani, "Least angle regression" *Annals of Statistics*,Vol. 32, pp. 407499, 2004.

[8] J.L. Starck, M. Elad, and D.L. Donoho, "Image decomposition via the combination of sparse representations and a variational approach", *IEEE Trans. On Image Processing*, Vol. 14, No. 10, pp. 1570-1582, October 2005.

[9] M. Elad, J-L. Starck, P. Querre, and D.L. Donoho, "Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA), *Journal on Applied and Computational Harmonic Analysis*, Vol. 19, pp. 340-358, November 2005.

[10] P.L. Combettes, V.R. Wajs, "Signal recovery by proximal forward-backward splitting," *Multiscale Model. Simul.*, Vol. 4(4),pp. 11681200,2005.

[11] D.L. Donoho, M. Elad, and V. Temlyakov, "Stable recovery of sparse overcomplete representations in the presence of noise", *IEEE Trans. Info. Theory*, Vol. 52, pp. 6-18, January 2006.

[12] J. A. Tropp, "Just relax: Convex programming methods for identifying sparse signals" *IEEE Trans. Info. Theory*, Vol. 51, num. 3, pp. 1030-1051, Mar. 2006.

[13] K. K. Herrity, A. C. Gilbert, and J. A. Tropp, "Sparse approximation via iterative thresholding,", *Proc. ICASSP*, Vol. 3, pp. 624-627, Toulouse, May 2006.

[14] D. L. Donoho, I. Drori, Y. Tsaig, J. L. Starck, "Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit", Stanford Technical Report, 2006.

[15] Joel A. Tropp, Anna C. Gilbert," Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Info. Theory* 53(12),pp. 4655-4666, 2007

[16] D. L. Donoho, Y. Tsaig, " Fast Solution of ℓ_1-norm Minimization Problems When the Solution May be Sparse", *IEEE Transactions on Information Theory*, November 2008.

[17] A. C. Gilbert, M. J. Strauss, J. A. Tropp, R. Vershynin, "One sketch for all: fast algorithms for compressed sensing," *STOC*,pp. 237-246, 2007.

[18] J. Bobin, J.L. Starck, J. Fadili, Y. Moudden, D. L. Donoho, "Morphological component analysis: An adaptive thresholding strategy," *IEEE Transactions on Image Processing*, Vol. 16(11), pp. 2675-2681, 2007.

[19] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, "A method for large-scale l1-regularized least squares," *IEEE Journal on Selected Topics in Signal Processing*,Vol. 1(4), pp. 606-617, December 2007.

[20] M. Elad, B. Matalon, J. Shtok, and M. Zibulevsky, "A Wide-Angle View at Iterated Shrinkage Algorithms", *proceedings of SPIE (Wavelet XII)* , San-Diego, August 26-29, 2007.

[21] M. Fornasier, H. Rauhut, "Iterative thresholding algorithms," *Applied and Computational Harmonic Analysis*, Vol. 25(2),pp. 187-208, 2008.

[22] T. Blumensath and M. Davies, "Iterative thresholding for sparse approximations," to appear in *Journal of Fourier Analysis and Applications, special issue on sparsity*, 2008.

[23] D. Needel, J. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," Accepted to *Appl. Comp. Harmonic Anal*, 2008.

[24] W. Dai, O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction", submitted to *IEEE Transactions on Information Theory*,2009.

[25] W. Yin, S. Osher, D. Goldfarb, J. Darbon, "Bregman iterative algorithms for ℓ_1-minimization with applications to compressed sensing," *SIAM Journal on Imaging Sci.*, Vol. 1(1),pp. 143168, 2008.

[26] R. Chartrand, "Exact reconstructions of sparse signals via nonconvex minimization," *IEEE Signal Process. Lett.*, Vol. 14, pp. 707-710, 2007.

[27] D. Donoho, "Compressed Sensing," *IEEE Transactions on Information Theory*, Vol. 52, pp. 489-509, April 2006.

[28] E. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. on Information Theory*,Vol. 52(2), pp. 489-509, February 2006.

[29] J. Bobin, J.-L. Starck, and R. Ottensamer,"Compressed sensing in astronomy" , *IEEE Journal of Selected Topics in Signal Processing* , Vol. 2, no. 5, pp. 718-726, 2008.

[30] D. L. Donoho, J. Tanner, "Neighborliness of randomly-projected simplices in high dimensions," *Proceedings of the National Academy of Sciences*, Vol. 102(27),pp. 9452-9457, 2005.

[31] David L. Donoho, "High-Dimensional centrally symmetric polytopes with neighborliness proportional to dimension," *Discrete and Computational Geometry*, Vol 35(4), pp. 617-652, 2006.

[32] D. L. Donoho, J. Tanner, "Counting faces of randomly projected polytopes when the projection radically lowers dimension," textitJ. Amer. Math. Soc., Vol. 22, pp. 1-53,2009.

[33] D. L. Donoho, J. Tanner, "Counting faces of randomly projected hypercubes and orthants, with applications," ArXiv.

[34] D. L. Donoho, J. Tanner, "Phase transitions as 'sparse sampling theorems' ," submitted to *IEEE Trans. on Information Theory*

[35] Jeffrey D. Blanchard, Coralia Cartis, Jared Tanner,"The restricted isometry property and l-q-regularization: Phase transitions for sparse approximation," submitted.

[36] R. Berinde, A. C. Gilbert, P. Indyk, H. Karloff, M. J. Strauss,"Combining geometry with combinatorics: a unified approach to sparse signal recovery", submitted 2008.

[37] `http://predictioncenter.org/.`

[38] `http://www.itl.nist.gov/iad/894.03/fing/fing.html`

[39] `http://www.frvt.org/FRGC/`

[40] K. Schnass, P. Vandergheynst, "Average performance analysis for thresholding," *IEEE Signal Processing Letters*, Vol. 14, No. 11, pp. 828-831, 2007.

[41] D. L. Donoho, A. Maleki, I. Ur Rahman, M. Shahram, V. Stodden, Reproducible Research in Computational Harmonic Analysis, *Computing in Science and Engineering*, v ol. 11, no. 1, pp. 8-18, Jan./Feb. 2009, doi:10.1109/MCSE.2009.15