A swarm based heuristic for sparse image recovery

Theofanis Apostolopoulos

King's College London, Department of Informatics
Strand, London, WC2R2LS, United Kingdom
`theofanis.apostolopoulos@kcl.ac.uk`

Abstract

This paper discusses the Compressive Sampling framework as an application for sparse representation (factorization) and recovery of images over an over-complete basis (dictionary). Compressive Sampling is a novel new area which asserts that one can recover images of interest, with much fewer measurements than were originally thought necessary, by searching for the sparsest representation of an image over an over-complete dictionary. This task is achieved by optimizing an objective function that includes two terms: one that measures the image reconstruction error and another that measures the sparsity level. We present and discuss a new swarm based heuristic for sparse image approximation using the Discrete Fourier Transform to enhance its level of sparsity. Our experimental results on reference images demonstrate the good performance of the proposed heuristic over other standard sparse recovery methods (L1-Magic and FOCUSS packages), in a noiseless environment using much fewer measurements. Finally, we discuss possible extensions of the heuristic in noisy environments and weakly sparse images as a realistic improvement with much higher applicability.

1998 ACM Subject Classification I.4.0 Image processing software

Keywords and phrases Compressive Sampling, sparse image recovery, non-linear programming, sparse representation, linear inverse problems

Digital Object Identifier 10.4230/OASIcs.ICCSS.2013.3

1 Introduction

The famous sampling theorem of Shannon-Nyquist has been very important in engineering. Straightforward and precise, it sets forth the number of measurements required to reconstruct any type of signal or image data. However, many real world applications, such as sound, images and video are represented, stored and processed in computers as big files or collections of bits, which has many disadvantages in comparison with small files; they require more storage space, they take longer to transmit and they demand an overwhelming computational cost for processing. For this purpose many signal/image compression techniques have been introduced including the emerging field of Compressed Sensing (CS). Compressive Sampling or Compressed Sensing (CS) is a fairly new area which was previously introduced empirically in the sciences (e.g. by Claerbout-Muir in Seismology) [3, 4, 5, 9]. CS as a cheap and fast sampling and recovery process has attracted considerable research with several new application areas over the past few years. By exploiting the image (sparsity) and the measurements (random samples) structure we are able to recover an image from what was previously considered as highly incomplete and inaccurate (under-sampled) measurements. Following the pioneer theoretical and practical works by Donoho [6], Candes, Romberg and Tao [4, 5, 9, 22] we are able to recover an under-sampled image, with high probability, by solving an ill-posed inverse problem, as a combinatorial optimization problem. Towards this direction, many variants and extensions of CS have been introduced in the literature recently (1000+ papers in the last 8 years) [21]. This paper proposes a new swarm based heuristic

Imperial College Computing Student Workshop 2013 (ICCSW'13).
Editors: Andrew V. Jones, Nicholas Ng; pp. 3–10
OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

for sparse image approximation and representation based on the key mathematical aspects of the CS method. We have already discussed and suggested the basics of this approach in signals [1, 17]. In this paper we aim to preset and extend the heuristic in images which introduces a more complete and realistic application. The heuristic is also compared with other well-known alternative methods in terms of recovery error, samples size and average computation time. The rest of this paper is organised as follows: The next Section presents the sparse image recovery problem. Then, we briefly discuss two well-known methods used for sparse image recovery (Section 3), while the proposed swarm-based iterative method is described in Section 4. Section 5 presents some experimental results of the proposed heuristic and its comparison with the other methods, while the Section 6 provides some conclusions and extensions of the proposed method.

2 Images as sparse representations

In computers, a image can be represented as a two dimensional array of points of the same size as the image. Each of these points is called pixel. Every pixel as a sample from the image and an element of its corresponding matrix represents the spatial irradiance distribution at the corresponding position. In other words, a pixel can be seen as a continuous function f of two variables m, n which correspond to its position of the array/grid (coordinates), while the function's value represents the type of light/color intensity. This light intensity value depends on the standard followed; it can be one value representing the tone of gray (gray level images) or multiple values for colour images, such as RGB and HSI pallets. For example, the corresponding array for a digital 512×512 gray level $2D$ image with 8 bit representation standard (256 colour intensity values) can be defined as [12, 18, 19, 22]:

$$f = \{f(m, n) = z; m, n = 0 : 511, z = 0 : 255\} \tag{1}$$

Sometimes to further enhance the processing steps or operations in an image (and thus its sparsity) we need to apply a so-called Unitary Transform [4, 12, 19, 22]. By this way we change the domain of representation (i.e. image function) from spatial (pixels) to frequency (spectra). In this case the image is represented as a linear combination of basis functions of a linear integral transform. This operation converts an image into one having relatively fewer values significantly different from zero. Obviously, the pursue of the best Transform domain which leads to the sparsest representation highly depends on the trade-off between the computation time and the size of the dictionary basis [12, 18, 19]. In this paper, we will apply the Discrete Fourier Transform (DFT), which uses cosines and sines as basis functions (i.e. $e^{i\omega} = \cos(\omega) + i\sin(\omega)$), to gray level images. The 2D DFT (spectrum) of an image $f(m, n)$ can be defined as [12, 18, 19, 22]:

$$X(u, v) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) \exp[-2\pi i(\frac{mu}{M} + \frac{nv}{N})], \tag{2}$$

for $u = 0, 1, \ldots, M-1$ and $v = 0, 1, \ldots, N-1$. The inverse DFT is given by:

$$f(m, n) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} X(u, v) \exp[-2\pi i(\frac{mu}{M} + \frac{nv}{N})], \tag{3}$$

for $m = 0, 1, \ldots, M-1$ and $n = 0, 1, \ldots, N-1$. Usually in images $M = N$, which is also the case for the test images discussed in this paper. Note also that there is one-to-one mapping between the spatial and frequency domain. The 2D DFT maps an $M \times N$ real-valued matrix

$f(m,n)$ on $M \times N$ complex-value matrix $X(u,v)$, while the inverse DFT maps the $X(u,v)$ on $f(m,n)$ [12, 18, 19]. In practice, the DFT is computed using the Fast Fourier Transform (FFT) algorithm, which is nothing but a computationally efficient way of obtaining the DFT coefficients based on the symmetries of the basis (matrix Ψ) [12, 18, 19, 22]. Many natural images have concise representations when expressed in a convenient basis. In CS we use DFT to enhance the sparsity of an image before down-sampling it. In image processing, for simplicity reasons (eg. Histogram of an image), it is very common to treat an $N \times N$ image as a $N := N^2$ vector and samples as a vector on the M frequencies ($M \ll N$); principle we will also adopt here. Let's assume we have a noiseless image $f \in \Re^N$ which we expand in an orthonormal basis (such as a Fourier basis) $\Psi = [\psi_1, \psi_2, \ldots, \psi_N]$ as $X = \sum_{i=1}^{N} f_i \psi_i$. Then the image can be represented as a sparse linear combination of atoms in Φ. In vector format, we have [3, 4, 5, 6, 12, 22]:

$$X_{N\times 1} = \Psi_{N\times N} f_{N\times 1}, \tag{4}$$

where Ψ is a unitary $N \times N$ matrix (basis) with $\psi_1, \ldots, \psi_N$ as columns and X is the vector of frequency coefficients with respect to Ψ. So, the N-point DFT is expressed as an N-by-N matrix multiplication, where f is the original input image and X is the DFT of the image. Then we can sense or collect partial information about X (measurements) as $y_k =< X, \phi_k >, k = 1, 2, ..., M$ or in vector format as [3, 4, 5, 6, 12, 22]:

$$Y_{M\times 1} = C_{M\times N} X_{N\times 1} = \Phi_{M\times N} \Psi_{N\times N} f_{N\times 1} \tag{5}$$

That is, we simply correlate the object we wish to acquire with the waveforms Φ, which is the measurement or sampling operator and Ψ is the sparcifying operator (Fourier transform) [3, 4, 5, 6, 8]. In fact, there is no formal difference between Φ and Ψ. In theory, the former refers to the dictionary of physical spectra and the later refers to the dictionary of image waveforms. In practice, Ψ is a partial Fourier matrix obtained by selecting M rows (i.e. measurements) uniformly at random, using Gaussian distribution, and then re-normalising the columns so that they are unit-normed (See [5, 6, 8, 12, 19, 22]). Note that the random Fourier ensemble is only used as a more realistic application and thus efficient recovery of the original image f still requires a unique sparsest solution. In a nutshell, the key steps of CS are: take the DFT of the desired image to enhance its sparsity, under-sample it (lossy compression) randomly, transmit/store it and then decompress (recover) it by solving an optimisation problem. As we will see in the next section, different recovery methods solve slightly different optimisation problems, though all these approaches serve the same purpose: the sparse recovery of a (compressed) image as a solution to an optimisation problem.

3 Methods for sparse recovery in images

CS is very advantageous in images which are sparse (have only a few non-zero entries) in a known basis provided that the measurements collected are incoherent (i.e. random) [3, 4, 5, 8, 12, 19, 22]. Since we are interested in sparsely representing highly under-sampled images, the linear system describing the measurements in (5) is under-determined and therefore has infinitely many solutions [3, 4, 5, 6]. This instance of an under-determined system of linear equations constitutes a linear inverse problem (LIP) [4, 5, 6, 8]. In this paper we will consider C as a random Fourier ensemble (rows are randomly chosen DFT vectors), in a noiseless environment. Note that randomness of sampling guarantees that we have a linearly independent system of equations and hence a unique solution. We will also restrict our approach to image restoration experiments applied to this problem and not to general applicability LIPs [4, 5, 8]. We discuss two well-known optimisation principles which are implemented in Matlab packages and have been extensively studied mathematically.

3.1 The L1 Magic

L1 MAGIC is a collection of MATLAB routines, based on standard interior-point methods, for solving optimization programs relevant to Compressive Sampling [5, 7]. In the case of the sparse image (noiseless) recovery problem the L1 Magic solves the TV minimisation problem with equality constraints which is a Second Order Cone Programming (SOCP) [4, 5, 7, 22]:

$$\min\ TV(X)\ \text{ s.t. }\ CX = Y, \tag{6}$$

where C is the Sampling/Sensing matrix (the under-sampling Fourier operator F_u), Y is the measurements vector, while TV stands for the Total Variation, which is the sum of magnitudes of the discrete gradient $D_{ij}X$ at every point/pixel x_{ij} of a FFT image X with i representing the rows and j representing the columns (assume sparsity in gradients) [4, 5, 7, 22]:

$$TV(X) := \sum_{ij}^{N-1} \sqrt{(x_{i+1,j} - x_{i,j})^2 + (x_{i,j+1} - x_{i,j})^2} = \sum_{ij} \|D_{ij}X\|_2 \tag{7}$$

The L1 Magic uses a log-barrier method to solve the SOCP in (6). It initially transforms it into a series of linearly constrained problems and then solves them by forming a a series of quadratic approximations (i.e. a Newtonian iteration step which proceeds by minimizing each of these systems of equations) [7].

3.2 The FOCUSS

FOCUSS package, which stands for FOCal Under-determined System Solver, is an algorithm designed to obtain sub-optimally sparse solutions to linear inverse problems in relatively noise-free environments [14, 20]. It is an affine-scaling transformation interior point optimisation algorithm which is based on conjugate gradient factorisation for finding sparse solutions of the following concave function [14, 20]:

$$\hat{X} = \text{ args } \min_X \frac{1}{2}\|Y - CX\|^2 + \lambda d_p(X), \tag{8}$$

where C is the Sampling/Sensing matrix, Y is the measurements vector, $0 < \lambda < 1$ is a regularisation parameter. This reflects the trade-off between the sparse residual $\|Y - C\hat{X}\|$ and the sparse source vector estimate $\hat{X}$ and depends on the compression rate of the sampled FFT image X. The quantity $d_p(X)$ corresponds to the following norm [14, 20]:

$$d_p(X) = \|X\|_{l_p} = \sum_{i,j} \|x_{ij}\|^p, \tag{9}$$

for $0 < p \leq 1$ which enforces sparse solutions to the problem.

4 Research Approach

A simple technique for recovering an image of interest X from partial measurements Y is to find a solution from an infinite set with the minimum sparsest norm [3, 4, 5, 6, 8, 9, 12, 19]:

$$\min\ \|X\|_{l_0}\ \text{ s.t. }\ Y = CX \tag{10}$$

where, the norm $\|.\|_{l_0}$ counts the non-zero elements of the vector X and thus $\|X\|_{l_0} = S$ for a S-sparse image (S non-zero entries). As X represents the partial Fourier measurements the image can be reconstructed as $f = \Psi X$. A common approach to overcome the difficulties of the combinatorial search required for solving (10) would be to replace it by its convex relaxation and particularly by substituting the l_1 norm for the l_0 pseudo-norm (For details see [3, 4, 5, 6, 9]). In this paper, we will follow a different approach which introduces an efficient way to approximate the l_0 by the following smoother, continuous and easier to differentiate Laplace function [1, 10, 13, 15, 16, 17]:

$$\|X\|_{l_0} \approx f(|X|,\sigma) = \sum_{i=1}^{N} 1 - f(|x_i|,\sigma) = N - \sum_{i=1}^{N} \exp(-\frac{|x_i|^2}{2\sigma^2}), \tag{11}$$

where x_i is the i-th element of vector X of length N and σ is a sequence index parameter. Among the advantages of this approach are the robustness of the l_0 norm to noisy samples, the number of measurements required, which is much smaller than the ones required by its convex analog (l_1 norm), and less restrictions in the design of Sensing matrices C. The problem in (10) is now reformed as an unconstrained optimisation problem:

$$\min \quad f(|X|,\sigma) = (M - \sum_{i=1}^{M} \exp(-\frac{(y_i - c_i x_i)^2}{2\sigma^2})) \tag{12}$$

where x_i and y_i are the i-th elements of vectors $X \in C^N$ and $Y \in C^N$ respectively, while c_i represents the i-th row of Sensing matrix $C \in C^{M\times N}$ with $M \ll N$. The purpose is to minimise both the objective function in (12) and the parameter σ. The value of this parameter represents the tradeoff between accuracy and smoothness of the approximation. The smaller the σ, the better the approximation, while the larger the σ, the smoother the approximation. In fact, the functional in (11) interpolates the function space between l_1 and l_0 across $\sigma \in [0,\infty)$ in the same manner as does l_p norm for $p \in [0,1]$ (i.e. $f(1,\sigma) = 1$ and $\sigma \to \infty$ admits $f(|X|,\sigma) \to \|X\|$). This approach has been successfully used for similar recovery problems and proven to yield a unique and sparse solution similar to the approach yielded by the l_0 quasi-norm (See [10, 13, 15, 16] for details and extensive results).

4.1 The Heuristic

The heuristic is an iterative stochastic swarm-based method for finding the global minimum of a non-convex, unconstrained continuous function in (12). The steps of the heuristic are:
Proposed l_0-norm based Heuristic:

```
Problem: Determine X ∈ C^N s.t. CX = Y.
Inputs: σ, C, Y, Iterations, Agents, Sparsity S, f(|X|,σ), Basis Ψ.
Outputs: best value f_*(|X_*|,σ), best sparse vector X_*, image f_*.
Main steps of the swarm based Heuristic:
Initial solution for every swarm i: X_i^(0) = ((C^T C)^(-1) C^T Y) + R^(0) × ||C^T Y||_∞
Set σ_i^(0) = ||Y − X_i^(0)||_∞ for every swarm i
  While (t < Iterations)
    For i := Agents/Swarms to 25 Do
           Evaluate f(|X|,σ) for every X_i^(t)
           Find current best X_*^(t) so as min f(|X|,σ)
           Set X_*^(t) = X_i'^(t) (keep the best i'th solution)
           Check X_*^(t) entries for feasibility
           Set all but S largest entries of X_*^(t) to zero
```

```
            Generate new solutions for all the other agents using (14)
     End For loop;
   Set σ^(t+1) = σ^(t) × 0.6
   End While loop;
Reconstruct image f_* = Ψ^{-1} X_* (IFFT of X_* ∈ C^N to derive f_* ∈ ℜ^N).
Display the recovered image f_*,Calculate the time and error recovery.
```

Initially the heuristic is initialised with a population of 25 agents each of which carrying a slightly different solution and σ parameter. A variation of pseudo-inverse $(((C^TC)^{-1}C^TY) + R^{(0)} \times \|C^TY\|_\infty)$ is chosen as an estimate of the initial sparse solution in (10), which will be further improved through the iterations of the heuristic. $X_i^{(t)}$ is the current solution vector for agent i at time t, while R is a vector of randomly generated values between 0 and 1, using the Normal distribution. Note that $\|.\|_\infty$ is the infinity or Chebychev norm, which is defined as $\|L\|_\infty = \max \|l_1\|, \ldots, \|l_N\|$ for a vector $L = [l_1, \ldots, l_N]$ in a finite dimensional coordinate space. Note that, at each iteration the current best solution $X_*^{(t)}$ is chosen after being corrected for sparsity and feasibility (be within the ranges of the original transformed image). Then a new solution is created for each remaining particle which is updated based on the following rule:

$$X_i^{(t)} = 2 \times R^{(t)} \times X_i^{(t-1)} + (1 - R^{(t)}) \times \sigma^{2L} \times L, \tag{13}$$

where, R is a vector of small random numbers, different for every swarm i, (t) is the current iteration, $X_i^{(t)}$ and $X_i^{(t-1)}$ is the current and the previously generated solution vector of swarm i and L is the infinity norm $\|C^T(CX_i^{(t-1)} - Y)\|_\infty$. The σ value is initially assigned to the maximum value between the samples vector and the sampled pseudo-inverse and then it is gradually decreased at each iteration. This assignment was chosen experimentally based on the nature of the initial vector. Note that due to the randomness in each step of the heuristic, there is no mathematical guarantee of achieving a global minimum as does its convex l_1 analogue. However, the local minimum found by solving the non-convex problem in (12) typically allows for accurate and successful recovery even at much higher under-sampling rates where linear optimisation fails (See Section (5) for details).

5 Simulations and Results

All the numerical experiments were performed on an Intel Core i5 CPU (3.20 GHz) with 3 GB RAM, using Matlab R2012b under MS Windows XP Pro. We have tested the performance of the heuristic as a sparse recovery method in two 256×256 images which have been extensively used for testing purposes, namely Shepp-Logan Phantom and Circles (See [2, 9, 13, 19, 21]). The result of the experiments is shown in Figure (1) which presents the original images, the Sampling pattern (i.e. number of lines through origin), the Back-projection (C^TY), which represents direct recovery from partial measurements, and the recovered image (estimate) using the methods L1 Magic, FOCUSS and the Heuristic. The performance of the heuristic is compared with the other methods in terms of recovery error (RE) as a metric to evaluate the recovered image quality. The recovery error was calculated as $\text{RE} = (\|\hat{X} - X\|_{l_2})/(\|X\|_{l_2})$, where $\hat{X}$ and X is the recovered and original image respectively, while the CPU cycles were used as a rough estimation of execution time for all the methods. The average time for the phantom image recovery was 468.27 ($\sim$ 10 mins) for L1 Magic with 15 Log-barier iterations, 322.85 ($\sim$ 6 mins) for FOCUSS with 15 iterations, $\lambda = 2.0e-3$ and $p = 0.5$, and 120 ($\sim$ 3 mins) for the heuristic with 23 iterations and 25 agents. The average time for the circle image recovery was 448 ($\sim$ 7 mins) for L1 Magic, 290 ($\sim$ 5 mins) for FOCUSS and 86 ($\sim$ 2

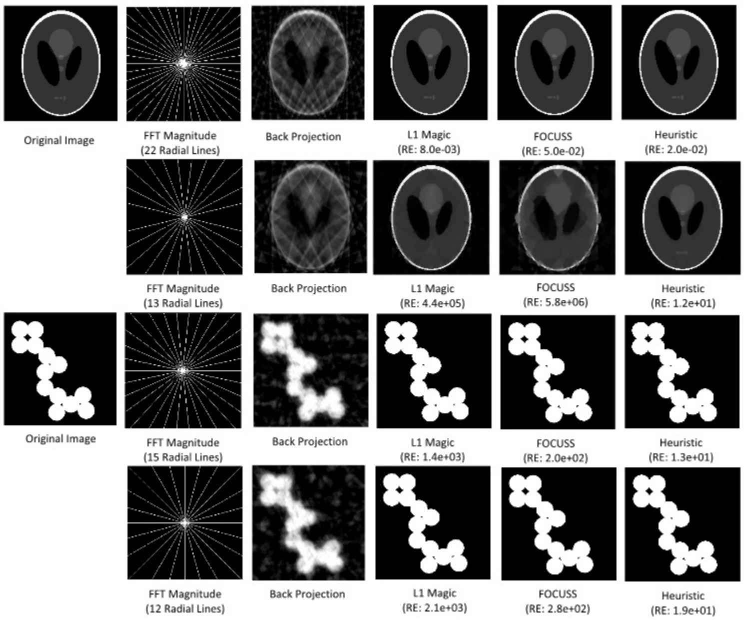

Figure 1 Image recovery experiments for L1 Magic, FOCUSS and Heuristic.

mins) for the heuristic, using the same parameters as previously. Notice that a few frequency coefficients (magnitudes) can capture most of the image energy, as most of such images are highly compressible. Notice also that the performance of the heuristic (in accordance with all the other methods) is increasing as the number of measurements increases and deteriorates as the sparsity of images decreases, which is expected as fewer measurements cause loss of image quality and thus loss of substantial information (i.e. aliasing in the reconstruction). However, the heuristic is found to have significantly better performance with smaller run times than the other recovery methods, particularly for much under-sampled data (less than 15 radial lines), while the difference between the recovered and the original image is hardly noticeable in some cases (particularly for more than 15 radial lines).

6 Conclusions

In this paper the performance of the proposed method for sparse image recovery was studied and compared with other methods. The heuristic essentially helps in faster and quicker sparse recovery of the test images by solving a non-convex unconstrained optimization problem with complex values, resulting in decreasing the requirement of the number of measurements needed by other alternative sparse recovery algorithms. It is expected that the performance of the heuristic, especially in noisy environments, can be improved by assigning weights to the objective function as an efficient way to improve the search direction. This approach has been proven to be useful and helps in better recovery for similar recovery problems and methods using the l_1, l_2 and l_0 norms (See [9, 11, 13]). Another possible direction is to investigate if we could design Sensing matrices which do not follow the Gaussian and Bernoulli distributions, or the random Fourier ensemble. These are the only distributions used to efficiently recover strictly sparse images from corrupted measurements (using the l_1 norm) as they satisfy the properties of UUP and RIP (i.e. mutual coherence between the Basis Ψ and the Sensing Φ matrices, for details See [4, 5, 8]). However, images of practical interest are generally weakly sparse or compressive, in essence that their sorted magnitudes in a known basis usually decay exponentially, and thus further experimentation may yield to reveal an even better recovery for weakly sparse images and signals (See [12, 19]).

Acknowledgements The author would like to thank his supervisor, Dr. Tomasz Radzik, for his insight and constructive comments and the anonymous reviewers for their suggestions.

References

1 T. Apostolopoulos. A heuristic for sparse signal reconstruction. In *ICCSW 2012*, volume 28 of *OASIcs*, pages 8–14. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2012.

2 Calibrated Imaging Lab at Carnegie Mellon University. Collection of test images. http://www.cs.cmu.edu/afs/cs/project/cil/ftp/html/v-images.html.

3 R. Baraniuk. Compressive sensing. *IEEE Signal Processing Magazine*, pages 118–120, 2007.

4 E. J. Candès. Compressive sampling. *Proceedings of the International Congress of Mathematicians, Madrid, Spain*, 2006.

5 E. J. Candès. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Trans on Information Theory*, 52(2):489–509, 2006.

6 D. Donoho. Compressed sensing. *IEEE Trans on Information Theory*, 52(4):1289–1306, 2006.

7 E. J. Candès et al. L1-magic: Recovery of sparse signals via convex programming. http://users.ece.gatech.edu/ justin/l1magic/downloads/l1magic.pdf.

8 E. J. Candès et al. Sparsity and incoherence in compressive sampling. *Inverse Problems*, 23(3):969–985, June 2007.

9 E.J. Candès et al. Enhancing sparsity by reweighted l_1 minimization. *Journal of Fourier Analysis and Applications*, 14(5):877–905, December 2004.

10 H. Mohimani et al. A fast approach for overcomplete sparse decomposition based on smoothed l_0 norm. *IEEE Trans on signal processing*, 57(1):289–301, November 2009.

11 J. K. Pant et al. Reconstruction of sparse signals by minimizing a re-weighted approximate l_0-norm in the null space of the measurement matrix. *Circuits and Systems,53rd IEEE International Midwest Symposium*, pages 430–433, August 2010.

12 J. L. Starck et al. *Sparse Image and Signal Processing; Wavelets, Curvelets, Morphological Diversity.* Cambridge University Press, UK, 2010.

13 J. Trzasko et al. Highly undersampled magnetic resonance image reconstruction via homotopic l_0-minimisation. *IEEE Trans. on Medical Imaging*, 28(1):106–121, January 2009.

14 K. Kreutz-Delgado et al. Dictionary learning algorithms for sparse representation. *Neural Computation*, 15(2):349–396, February 2003.

15 P. Huber et al. *Robust Statistics.* Wiley, 2 edition, 2009.

16 S. Ashkiani et al. Error correction via smoothed l_0-norm recovery. *IEEE Statistical Signal Processing Workshop (SSP)*, pages 289–292, June 2011.

17 T. Apostolopoulos et al. A swarm based method for sparse signal recovery. In *ARSR/SWICOM 2013, Luton*, pages 1–5, 2013.

18 B. Jaehne. *Digital Image Processing.* Springer, 2002.

19 S. Mallat. *A Wavelet Tour of Signal Processing.* Academic Press, 3 edition, 2009.

20 J. F. Murray. Focuss website. http://dsp.ucsd.edu/ jfmurray/software.htm.

21 Compressive Sensing Resources. Rice university. http://dsp.rice.edu/cs.

22 J. Romberg. Imaging via compressive sampling. *Signal Processing Magazine, IEEE*, 25(2):14–20, March 2008.