A Micro-Mirror Array based System for Compressive Sensing of Hyperspectral Data

Yehuda Pfeffer
Dept. of Electrical Engineering
Technion - Israel Institute of Technology

Michael Zibulevsky
Dept. of Computer Science
Technion - Israel Institute of Technology

January 18, 2010

Abstract

We introduce a system for hyperspectral imaging, which includes a micro-mirror array that projects subsets of image pixels onto a prism (or diffraction grating), followed by a CCD-type sensor. This system allows generalized sampling schemes termed as Compressed Sensing (CS). We acquire only a fraction of the samples that are required to obtain the full-resolution signal (hyperspectral cube in our case), and by means of non-linear optimization recover the underlying signal. We use a prior knowledge about the signal sparsity in some fixed dictionary, and also its limited total variation. In the practical setting developed here, the feasible sampling is not ideal for CS due to practical limitations, and the sensed signal does not necessarily meet the strict sparsity demands of CS theory. Therefore we introduce additional measurements of full-resolution image using small number of filters similar to RGB. As a result, we obtain a feasible system for hyperspectral imaging that enables faster acquisition compared to traditional sampling systems.

1 Introduction

Acquisition of hyper-spectral (HS) data takes some time. Not all the data cube is acquired at once - it is taken either line by line, or by sequentially sampling the fourier domain (See subsubsection 3.2.1 for details). The long acquisition time limits the usability of HS imaging - moving object, for example cannot be captured properly by HS camera compared to color imaging. There is also an issue of the large amount of data involved in HS imaging when that data needs to be transmitted. Compression algorithms solve the data handling problem successfully, but the lengthy acquisition time cannot be solved by compression algorithms such as JPEG [41], since JPEG needs all the data in order to do the compression. Thus, acquisition time can be shortened in two ways: First, by reducing the acquisition time of each line. This will result in lower SNR if the illumination intensity
will not increase accordingly, which is the case when we use natural light. The second way of reducing acquisition time is by reducing the spatial and/or spectral resolution of the HS cube, obviously resulting in undesirable lower resolution data.

In this report we will try to use ideas from the field of Compressive Sampling, also termed Compressed Sensing (CS) [1], [3]. By using CS we follow the second approach - sampling less data in order to achieve faster acquisition and compression. The CS theory tells us that for many signals we can sub-sample the signal and still be able to reconstruct the original signal with good accuracy. The signal has to be either sparse or its representation in some fixed dictionary be sparse. We term both cases as sparse signal. The compressed sampling must be done under certain condition, and the reconstruction algorithm is based on nonlinear optimization. The details will be given in subsection 2.2. Compressed sensing is useful when we have some strong limitations on the amount of data that we can acquire. Examples include MRI [20], where the signal acquisition speed is limited by physical and physiological constraints, analog-to-digital [19] converters where technology and hardware costs limits the maximum rate, and medical imaging techniques involving ionizing radiation such as CT and PET, in which we are interested in minimizing the amount of radiation.

In the Methods section we will display the main theoretical results in CS, and then demonstrate its use on one dimension signals. We will also experiment using CS for 2D image acquisition in subsection 2.3, both on synthetic data and real-life images. The use of CS on images is not straightforward, and this subsection expands both acquisition and the reconstruction schemes beyond the classic results. In the Compressed sensing of hyperspectral data section we will develop a novel sampling system for hyper-spectral data that is based on compressive sensing. We will simulate its use, and develop novel reconstruction schemes that will take advantage on the special structure of the HS data cube.

2 Methods

2.1 Compressive Sensing - Introduction

Compressive Sensing (CS), [1], [3] is an emerging field that is based on the understanding that for certain types of signals, it is possible and practical to sample $M \ll N$ samples, were $N$ is the signal length for discrete data, or the minimal number of samples required by the Nyquist theorem. Throughout this report the signal $x \in R^N$ is sampled by the sampling (or sensing) matrix $\Phi \in R^{M \times N}$, obtaining the measurement vector $y = \Phi x$. The classic results, [4], [5], are established for sparse or compressible signals - the sampled signal $x$ must be either sparse or compressible w.r.t a known dictionary. However, [10] gives a more general description of the idea behind CS: if the under-sampled signal $x$ lies in a low-dimensional non-linear manifold, then it is possible to reconstruct $x$ from the $M$ mea-
measurement vector $y$ assuming the sampling is done under certain limitations. If the non-linear manifold is the set of sparse signals (possibly in some fixed and known dictionary), then there are well defined limitations on the sampling matrix [6], [14]. If, however, the non-linear manifold is the set of signals that have low Total-Variation (TV, [13]) as in [18], then there are no theoretical results that justify the success of CS, although there are practical results. Other such restrictions can be imposed on the reconstruction process, restrictions that are specific to the sampled signal. In this work we focus on efficient ways to sample and reconstruct hyperspectral (HS) data. The special structure of the HS data, along with a novel sampling scheme allow us to achieve surprisingly good results on data that does not meet the theoretical sparsity demands under the current CS framework.

2.2 Compressive Sensing - Main results

Some definitions: Let $x \in \mathbb{R}^N$ be the signal to be sampled. Let $\Phi \in \mathbb{R}^{M \times N}$ be the sampling matrix, and let $y = \Phi x$, $y \in \mathbb{R}^{M}$ be the measurement vector. We consider the noise-free case just defined, as well as noisy measurement vector: $y = \Phi x + n$, were $n \in \mathbb{R}^M$ is the vector of measurement noise.

Sparsity and compressibility: an $s$-sparse signal $x$ has no more than $s$ entries different from 0. Formally: $\|x\|_0 \leq s$, were $\| \cdot \|_0$ is the $\ell_0$ pseudo-norm, which counts the non-zero entries of a vector. An $s$-compressible signal is a signal that most of its energy is concentrated around no more than $s$ entries. A sparse approximation to $x$, $x_s$, is the closest $s$-sparse vector to $x$ under the $\ell_2$ norm. $x_s$ is obtained from $x$ by setting all the entries in $x$ that are not among the $s$ largest to zero.

Perfect reconstruction of sparse signals: given that $x$ is $s$-sparse, it is possible to reconstruct $x$ if $\Phi$ is such that every $2s$ columns of $\Phi$ are linearly independent. The reconstruction is done by solving:

$$\min_x \{\|y - \Phi x\| \text{ s.t. } \|x\|_0 \leq s\} \quad (1)$$

Direct solution of (1) requires exhaustive search over all the $s$-sparse $N$ over $s$ many combinations, which is not a very practical reconstruction method - efficient schemes will be presented later. The solution of (1) is covered in the field of sparse approximations [14]. The perfect reconstruction property is easy enough to prove [7]: if there are two different solutions to (1), $x_1 \neq x_2$, were both $x_1$ and $x_2$ are $s$-sparse, then $\tilde{x} = x_1 - x_2$ is $2s$ sparse and $\Phi \tilde{x} = 0$. This means that there are $2s$ (or less) columns of $\Phi$ that are linearly dependent, in contradiction to the assumption. $s$ columns of a Random measuring matrices are linearly independent for any $s \leq N$ with probability one - the probability that any sub-matrix will have an exactly zero eigenvalue is zero. However, ill-conditioned sub-matrices will not allow reconstruction in any practical setting. Also, the solution of (1) is not feasible for large problems. Practical reconstruction schemes, compressible (and not strictly sparse) signals and noisy measurements require stronger conditions on $\Phi$. It was shown in [8], [14] that solving the
convex $\ell_1$ optimization problem,

$$\min_x \{ \|x\|_1 \quad s.t. \quad y = \Phi x \}$$  \hspace{1cm} (2)$$

will recover $x$, given that $x$ is sparse and that $\Phi$ obeys the Restricted Isometry Property displayed below w.r.t to $x$’s sparsity.

### 2.2.1 Dictionaries

Most of the interesting signals are not sparse or even compressible in the signal’s domain (time or space), but have some sparse representation in some fixed dictionary, for example wavelets [24], [39]. A dictionary is a set of $K$ atoms, $a_i \in \mathbb{R}^N$, $i = 1, \ldots, K$. A dictionary can be an orthogonal basis, such as wavelets, Fourier transform or Discrete Cosine Transform (DCT) (in which case $K = N$), or an overcomplete dictionary when $K > N$, for example wavelets packet, undecimated wavelets. Overcomplete dictionaries have greater “sparsifying” abilities [14]. Let $\Psi$ be the dictionary matrix with the dictionary atoms as its columns. In the case of specific orthogonal transform, $\Psi$ is the inverse transform matrix. Sparsity in a known dictionary can be easily incorporated into the CS scheme [21]: Assume that vector $z$ is the sparse coefficients of $x$’s representation in the domain of $\Psi$: $x = \Psi z$, and $z$ is sparse. Then, the measurements are obtained by $y = \Phi \Psi z$. Define: $A = \Phi \Psi$. $A$ will take the place of $\Phi$, and $z$ will take the place of $x$ in (1) and (2). There is, however, a fundamental difference: the aforementioned properties of $\Phi$ must now apply to $A$. In the following sub-sections, the requirements for $\Phi$ will be defined formally, and we shall see that while there are well known results for random $\Phi$, the addition of the dictionary $\Psi$ can result in an $A$ matrix having very different properties - especially when using non-orthogonal overcomplete dictionaries.

In order to proceed, the notions of Restricted Isometry, Coherence and Babel function will now be defined.

### 2.2.2 Restricted Isometry Property (RIP)

The requirement on $\Phi$ are formalized through the idea of the Restricted Isometry Property (RIP), which is a way to quantify the “orthogonality” of any sub-matrix of $\Phi$.

**Definition** [7], [22]: We say that a matrix $\Phi$ satisfies the Restricted Isometry Property of order $s \in \mathbb{N}$, $s < N$, if there exist an Isometry Constant, $0 < \delta_s < 1$ such that

$$(1 - \delta_s) \|x\|_2^2 \leq \|\Phi x\|_2^2 \leq (1 + \delta_s) \|x\|_2^2$$  \hspace{1cm} (3)$$

holds for all $s$-sparse vectors. $\delta_s$ will be the smallest number that satisfies (3). Orthogonal matrices have $\delta_s = 0$ for all $s$. $\delta_s < 1$ allows for reconstruction for any signal $x$, as explained in the previous subsection. Stricter conditions on $\delta_s$ are required for practical recovery using (2). The isometry constant $\delta_s$ of a given matrix is hard to find, but there are known results for $\delta_s$ of random matrices.
2.2.3 Restricted isometry - Example

Gaussian random matrix: Let $\Phi$ be an $M \times N$ random matrix, who's entries are iid gaussian random variables with zero mean and variance $1/M$:

$$\Phi_{i,j} \sim N(0, \frac{1}{M})$$

Bernoulli random matrix: Let $\Phi$ be an $M \times N$ random matrix, who's entries are iid $\pm \alpha$ Bernoulli random variables:

$$\Phi_{i,j} = \begin{cases} \frac{1}{\sqrt{M}} \text{ w.p. } \frac{1}{2} \\ -\frac{1}{\sqrt{M}} \text{ w.p. } \frac{1}{2} \end{cases}$$

Lemma 3.1 in [8] establishes that: Let $r = s/N$, and $H$ be the entropy function:

$$H(q) = -q \log q - (1-q) \log(1-q), \quad 0 < q < 1$$

And let

$$f(r) = \sqrt{N/M} \cdot \left( \sqrt{r} + 2H(r) \right)$$

For each $\epsilon > 0$ the restricted isometry constant $\delta_s$ of a gaussian matrix obeys:

$$P\left(\delta_s > \left[ 1 + (1 + \epsilon) f(r) \right]^2 - 1 \right) \leq 2e^{-\epsilon NH(r)/2} \quad (4)$$

For the Gaussian case. The proof is based on deviation bounds of gaussian random matrices' extremal singular values.

A recent proof of the RIP based on the Johnson-Lindenstrauss lemma in [22] gives us a different result: Follow the notation from the last result, and choose the lowest $c_1$ such that: $s \leq c_1 M / \log(N/s)$. Define

$$c_2 = c_0 \left( \frac{\delta}{2} \right) - c_1 \left[ 1 + \frac{1 + \log(12/\delta_s)}{\log(N/s)} \right]$$

were we use

$$c_0(\epsilon) = \frac{\epsilon^2}{4} + \frac{\epsilon^3}{6}$$

The RIP with isometry constant $\delta_s$ will hold with probability $\geq 1 - e^{-c_2 M}$. This result is valid for both the Gaussian and the Bernoulli ensemble due to the fact that the proof is based on the concentration inequality [22], which holds in both cases (and other cases as well).

Lets look on a numerical example: let $N = 21000$, $M = 7000$, $s = 6$. We also set $\delta_s = 1 - \sqrt{2}$ - this specific value will become important later. The RIP holds w.p. $1 - 4.3 \cdot 10^{-6}$, according to the first result, and w.p. of $1 - 3.5 \cdot 10^{-5}$ according to the second results (the highest one is closer to the truth, as both results are correct). While the second result gives tighter bound on the probability of success, the first result may hold when the second one is meaningless, for example $N = 18000$, $M = 6000$, $s = 6$, were we get for the first result that the RIP holds w.p. $1 - 0.044$, and a meaningless number in the second result (a negative $c_2$).

The compression rate $M/N$ in the example is $1/3$, but for the RIP to (provably) hold for such compression rate, the dimensionality of the problem must be very large and the signal must be very sparse - here $s = 6$, so
$x$ will have to be $s/2 = 3$ sparse for provable reconstruction by $\ell_1$ methods [7]. Most signals are not sparse enough to meet the demands above. However, numerical experiments that will be displayed in subsection 2.2.7 demonstrate that CS reconstruction schemes give good results for practical synthetic and real-world problems. Two explanations for this are: 1) The proofs of the RIP are based on proving that (3) holds for a certain combination of $s$ columns of $\Phi$, and then using the union bound to prove the RIP on all possible combinations, which might be very loose. 2) There are many ($N$ over $s$) combinations, and even if for some of them (3) does not hold, the success probability is still very high. This means that even if the RIP doesn’t hold in general, for most cases we still get good results.

2.2.4 Coherence and the Babel function

The coherence [11], [14], of a dictionary $A \in R^{N \times K}$ with unit-norm columns $\{a_j\}_{j=1}^K$ is defined as

$$\mu = \max_{k \neq l} |\langle a_k, a_l \rangle| = \max_{i \neq j} |(A^* A)_{i,j}|$$

The coherence of orthogonal matrices is of course 0. For general overcomplete dictionaries, a lower bound on the coherence is

$$\mu \geq \sqrt{\frac{K - N}{N(K - 1)}}$$

A union of up to $N + 1$ concatenated orthonormal bases (multi-ONB) has coherence $\mu \geq N^{-1/2}$, and the equality is attainable.

While the coherence is easy enough to calculate, it is a blunt instrument - it only reflects the external correlation of the dictionary. A more delicate tool is the Babel function, [14]:

$$\mu_1(s) = \max_{k \in \Lambda} \max_{|\Lambda| = s} \sum_{k \in \Lambda} |\langle a_j, a_k \rangle|$$

And the following connection holds: $\mu_1(s) \leq s \mu$. When a dictionary have an analytic description, it might be possible to calculate its Babel function.

We now have the mathematical tools needed to discuss the reconstruction algorithms under the CS scheme.

2.2.5 Reconstruction - greedy algorithms

The solution of (1) can be approximated by using sub-optimal greedy algorithms, most notably Matching Pursuit (MP) and Orthogonal Matching Pursuit (OMP) [11], [14]. A novel reconstruction algorithm dedicated to CS reconstruction is Tropp’s CoSaMP [7]. Tropp shows in [14] that OMP will solve (1) correctly whenever $x$ is exactly $s$-sparse and

$$\mu_1(s) + \mu_1(s) < 1$$
In fact, Basis Pursuit (BP, (2)) will also give the correct solution to (1) under the same condition (again, [14]). In terms of coherence, OMP (and BP) is correct whenever

\[ s \leq \frac{1}{2} (\mu^{-1} + 1) \]

When \( x \) is not sparse, the sparse approximation can be recovered if \( \mu_1(s) < \frac{1}{2} \).  

\[ \| x - x_{s,OMP} \| \leq \sqrt{1 + C} \| x - x_s \| \]

were \( x_{s,OMP} \) is the result of the OMP algorithm after \( s \) steps, and

\[ C < \frac{s(1 - \mu_1(s))}{1 - 2\mu_1(s)} \]

**CoSaMP algorithm**: CoSaMP [?] is a greedy algorithm using the RIP property of matrix \( \Phi \): while \( \Phi^* \Phi \) is not equal to the unity matrix, it is not so far from it (the diagonal elements are 1’s, and the off-diagonal elements are smaller than the coherence \( \mu \)). Thus, \( \tilde{x} = \Phi^* \Phi x \) (which is obtained by \( \tilde{x} = \Phi^* y \)) is a reasonable approximation. Similarly to OMP, CoSaMP iteratively improves the reconstruction, working on the residual on each step. CoSaMP requires \( \delta_{4s} \leq 0.1 \), or \( \delta_{2s} \leq 0.025 \) to provably achieve accurate reconstruction.

### 2.2.6 Recovery by \( \ell_1 \) minimization

Two recent results [7] bound the reconstruction error obtained by solving an \( \ell_1 \) optimization problem for 1) the noiseless case, and 2) the noisy case. Those results also unify the treatment of exactly sparse signals and compressible signals. Noiseless recovery: Assume that the Isometry Constant of \( \Phi \), \( \delta_{2s} \) obeys \( \delta_{2s} < \sqrt{2} - 1 \). Let \( x^* \) be the solution to (2). Then \( x^* \) obeys:

\[ \| x^* - x \|_1 \leq C_0 \| x^* - x_s \|_1 \]  
\[ \| x^* - x \|_2 \leq C_0 s^{-1/2} \| x^* - x_s \|_1 \]

\( C_0 \) is given explicitly below. (6) means that for exactly sparse signals, exact recovery is promised. This is a dramatic improvement over previous results [5]. Noisy recovery: In the noisy recovery setting we have \( y = \Phi x + n \), were \( n \) is the noise vector and \( \| n \| \leq \epsilon \). (2) will be replaced by:

\[ \min_{x} \{ \| x \|_1 \quad s.t. \quad \| y - \Phi x \|_2 \leq \epsilon \} \]

Let \( x^* \) be the solution of (7). If \( \delta_{2s} < \sqrt{2} - 1 \) and \( \| n \| \leq \epsilon \), \( x^* \) obeys:

\[ \| x^* - x \|_2 \leq C_0 s^{-1/2} \| x^* - x_s \|_1 + C_1 \epsilon \]

\( C_0 \) and \( C_1 \) constants: let

\[ \alpha = \frac{2(1 + \delta_{2s})}{1 - \delta_{2s}}, \quad \rho = \frac{\sqrt{2}\delta_{2s}}{1 - \delta_{2s}} \]

\[ C_0 = \frac{2(1 + \rho)}{1 - \rho}, \quad C_1 = \frac{2\alpha}{1 - \rho} \]

7
2.2.7 Examples - synthetic sparse signals

Let us consider the following example: \(N = 512\), \(M = 128\). The signal \(x \in \mathbb{R}^N\) in this example is \(s\)-sparse, with its non-zero elements chosen at random uniformly. The value of the non-zero elements is realized from the Laplacian distribution, and its absolute value is limited to be higher than 1. The purpose of the experiments is to demonstrate and test different reconstruction methods and different sampling schemes. The sampling scheme and reconstruction method will be tested on signals having variable sparsity and noise level.

**Different reconstruction schemes** The first experiment tests the performance of the reconstruction method - OMP, CoSaMP and convex relaxation (\(\ell_1\)). Following [5] and the simulations in [34], the sampling matrix, \(\Phi \in \mathbb{R}^{M \times N}\) is a random gaussian matrix who’s rows have been orthogonalized. Different levels of noise, as well as different levels of sparsity (\(s \in \{5, 10, 15, 20, 25, 30, 35, 40, 50\}\)) are tested. The results of the \(\ell_1\) reconstruction are displayed in figure 2.2.7 a. The results of the OMP reconstruction are displayed in figure 2.2.7 b. The average run time of a single \(\ell_1\) reconstruction is 2 seconds, while the run time of OMP is 0.05 seconds. The qualities of \(\ell_1\) and OMP reconstructions are comparable, with OMP better than \(\ell_1\). This a surprising result, as \(\ell_1\) considers better than OMP. The reason for the apparent superiority of OMP over \(\ell_1\) might be the specific setting of the problem, or incomplete convergence of the \(\ell_1\) optimization. Either way, the methods seems at least comparable, and OMP is about 40 times faster, does not require parameters calibration and is very simple to implement. The next experiments will be based on OMP. Still, when the time will come for solving the hyperspectral reconstruction problem, the \(\ell_1\) method will be tested again. We learn from the graphs that when using OMP reconstruction, a signal having up to 25 non-zero entries will be reconstructed perfectly given the noise STD is 0.05. This is much better compared to the theoretical bounds.

**Different sampling matrices** Here we test the influence of the sampling scheme on the performance of the reconstruction. We test random gaussian, orthogonalized gaussian, random \(\pm 1\) and structured \(\pm 1\) matrices. We also test a 0 – 1 version of the \(\pm 1\) matrices. All the sampling matrices rows have unit norm. The signals are realized as in the previous subsection, and the noise STD is 0.025. OMP is used for reconstruction. The reconstruction performance are displayed in figure 2.2.7. The performance of the 0 – 1 versions of the \(\pm 1\) matrices are not displayed because they were consistently worse than their equivalent \(\pm 1\) matrices. That phenomenon is due to the multiplication of a matrix with sparse vector - the inner product of a row with half of its entries set to zero with a random sparse vector is likely to be small, resulting in smaller measurement vector \(y\), which makes the noise more prominent.

By inspection of figure 2.2.7, we come to the conclusion that structured sampling matrices perform better than random matrices, both in the Gaussian case and in the binary case. This is coherent in a way with [10], although we use different structuring. Gaussian and binary
Figure 1: CS of synthetic sparse data, $\ell_1$ (a) and OMP (b) reconstructions. Each line stands for different noise level. The x-axis indicates the sparsity of the signals, and the y-axis indicates the probability of success in CS reconstruction. Success is defined as identifying the correct $s$ non-zero entries of the signal.
Figure 2: CS of synthetic sparse data, OMP reconstruction. Each line stands for different sampling matrix. The x-axis indicates the sparsity of the signals, and the y-axis indicates the probability of success in CS reconstruction. Success is defined as identifying the correct $s$ non-zero entries of the signal.
matrices have similar performance. It is important to note that when
dictionaries will be involved, the results may change.

2.2.8 Overcomplete dictionaries and CS

When we want to use CS for signals that are sparse in some dictionary 
(possibly overcomplete), we would like to know what happens to the RIP 
of the composed matrix $A = \Phi \Psi$. [21] gives an answer that covers both the 
orthogonal basis and the overcomplete dictionary cases. By analyzing the 
way random matrices operate on vectors according to the concentration 
inequality, theorem 2.2 in [21] establishes that: Let $\Phi \in R^{M \times N}$ be a sam-
pling matrix satisfying the concentration inequality, and let $\Psi \in R^{N \times K}$ 
be a dictionary with restricted isometry $\delta_s(\Psi)$. Assume that

$$M \geq C\delta^{-2} \left( s \log (K/s) + \log (2e(1 + 12/\delta)) + t \right)$$

for some $0 < \delta < 1$ and $t > 0$. Then with probability $1 - e^{-t}$ the composed 
matrix $A = \Phi \Psi$ has restricted isometry constant $\delta_s(\Phi \Psi) \leq \delta_s(\Psi) + \delta (1 + \delta_s(\Psi))$

Were $C \leq 23.143$ for both the Gaussian and Bernoulli sampling matrices. 
When $\Psi$ is an orthogonal matrix, we have $\delta_s(\Psi) = 0$ and we get the 
familiar result from [22]. In terms of coherence and babel function, Lemma 2.3 in [21]gives us the following connection for any dictionary:

$$\delta_s \leq \mu_1(s - 1) \leq (s - 1)\mu$$

2.2.9 Examples - CS with different dictionaries

We repeat the experiments from subsubsection 2.2.7, with signals that are 
sparse in different dictionaries or bases. Denote $\Phi$ as the sampling matrix, 
and $\Psi$ as the inverse of the sparsifying transform, or as the dictionary - the columns are the unit norm atoms. Since the RIP of the composite matrix 
$A = \Phi \Psi$ determine the probability of successful CS reconstruction, we 
cannot assume that a sampling matrix that worked well in the experiments 
on sparse signals will perform well here. Again, we use OMP and set 
the noise STD to 0.025. The $x$ signals will be realized as follows: $z$, the 
coefficients vector is realized exactly as $x$ is realized in subsubsection 2.2.7, 
and $x = \Psi z$. In the first stage $\Psi$ will be orthogonal (or bi-orthogonal), 
and later we will also examine overcomplete dictionaries.

Orthogonal bases The first set of dictionaries tested were orthogonal 
bases - db1 (Haar), db2 and Bi-Orthogonal 9-7 wavelets, and DCT base. 
The results are displayed in figure 3 in subplots a, b, c and d accordingly.

Indeed, the results are different when using dictionaries - in the DCT 
and Haar WT cases, the Hadamard based sampling matrices show very 
weak performance. This means that if the signals are sparse under either 
the DCT or Haar transforms, a Hadamard based sampling matrix cannot 
be used. The last result is not surprising, as we already know that the 
sampling matrix and the dictionary should be as incoherent as possible, 
and random rows from the Hadamard matrix are similar to columns from
Figure 3: CS of synthetic sparse data, different bases: db1 (Haar) wavelets (a), db2 wavelets (b), Bi-Orthogonal 9-7 wavelets (c), DCT base (d). OMP reconstruction. Each line stands for different sampling matrix. The x-axis indicates the sparsity of the signals, and the y-axis indicates the probability of success in CS reconstruction. Success is defined as identifying the correct $s$ non-zero entries of the signal.
Figure 4: Coherence of the composite matrix $\Phi \Psi$, for different sampling matrices and different bases. Each group of bars stands for a specific sampling matrix, see legends.
the inverse Haar transformation matrix in their 0 – 1 structure, and to columns from the inverse DCT transformation matrix in their oscillatory structure. A detailed comparison of coherence of $\Phi \Psi$ can be found in figure 4. When $\Psi$ is the identity matrix, the coherence of $\Phi$ is calculated. The combinations of the different bases with the Hadamard based sampling matrix exhibit high coherence, although the coherence of the Hadamard sampling matrices themselves is the lowest. The other sampling matrices - Gaussians, Rademacher exhibit the same coherence with or without the bases. The general conclusion from the last experiment is that it is better to use a randomized matrix rather than a structured one. This is because a random matrix is not likely to be similar to a dictionary, as dictionaries and bases are usually structured. Moreover, while the sampling of the signal is done once and the sampling matrix cannot be changed afterwards, the dictionary used for reconstruction does not have to be chosen in advance. It is possible to try several dictionaries for reconstruction until the result is satisfactory. Therefore, it is advantageous to use a sampling matrix that will allow the use of as many dictionaries as possible, and this is why random sampling matrices are best. Of course, if we have a system designed for a specific type of signal, a signal that is sparse w.r.t a known dictionary, we should use a sampling matrix that performs best with that base, See [10].

Overcomplete dictionaries Now we use two overcomplete dictionaries: the first is a combination of two orthogonal transforms - db2 wavelets and DCT. The second is dual-tree complex wavelets [35]. The results are displayed in figure 2.2.9, and the coherence is displayed in figure 2.2.9. The reconstruction results are considerably worse than when using orthogonal bases for the same sparsity. The coherence graph also shows that the $\Phi \Psi$ matrices are more coherent than before. Still, the results for the WT db2 + DCT dictionary are reasonable, and one should keep in mind that in many cases, a signal does not have sparse representation in an orthogonal basis, and does have a sparse representation in an overcomplete dictionary [23]. Hence, overcomplete dictionaries will be used further in this work.

2.3 Imaging and compressed sensing

Before we describe the main part of this work, compressive sensing of hyperspectral data, let us examine a simpler problem, the problem of CS of regular images (grey scale). In some cases it makes sense to use CS for imaging - for example in such cases when an imager that works in some special wavelength is very expensive or doesn’t exist. See for example [18]. The sensing part is implemented using a Micro-Mirror Array (MMA). The reconstruction requires the ability to calculate $\Phi x$ in an efficient way - even for a moderate resolution image of $256 \times 256$, $\Phi$ is a 16,384 \times 65,536 matrix - too big to use or store. Two ways to reduce the sampling complexity are first: to use a randomly chosen coefficients of a Fourier transform of the signal [3], which is naturally extended to any linear transformation. The second way is to use sparse sampling matrices [16]. Those are rather strict limitations - while the mirror array can realize a general sampling
Figure 5: CS of synthetic sparse data, different dictionaries: db2 wavelets and DCT (a), and dual-tree complex wavelets (b). OMP reconstruction. Each line stands for different sampling matrix. The x-axis indicates the sparsity of the signals, and the y-axis indicates the probability of success in CS reconstruction. Success is defined as identifying the correct $s$ non-zero entries of the signal.
Figure 6: Coherence of the composite matrix $\Phi \Psi$, for different sampling matrices and different dictionaries. Each group of bars stands for a specific sampling matrix, see legends.
matrix, it allows a \{0,1\} sampling matrix more naturally. The important case of \{0,1\} sampling matrix doesn’t fit into most of the common linear transforms, and we saw in 2.2.7 that the use of a sub-sampled Hadamard transform sampling matrix results in worse reconstruction performance, compared to a random \pm 1 matrix. In this subsection we present a novel sampling scheme for 2D data, a scheme that is more general than using fast implemented linear transforms, but nearly as efficient in many cases.

2.3.1 Separable sampling matrix

Let $X \in \mathbb{R}^{\sqrt{N} \times \sqrt{N}}$ be our $N$ elements image, and let $x \in \mathbb{R}^N$ be the column stacking of $X$. A compressed sampling of $x$ by a $\Phi \in \mathbb{R}^{M \times N}$ sampling matrix with $M = N/\alpha$ ($\alpha$ is the compression ratio) requires $N^2/\alpha$ multiplications. Let $\Phi_1$ and $\Phi_2$ be such that $\Phi = \Phi_1 \otimes \Phi_2$, and $\Phi$ is a subset $I_M$ of $M$ rows from $\Phi$. Let $Y = \Phi_1 X \Phi_2$, and let $y$ be the column stacking of $Y$, and $y = \Phi x$. Then we have $\tilde{y}_{I_M} = y$. The complexity of calculating $Y = \Phi_1 X \Phi_2$: $\tilde{\Phi}_1, \tilde{\Phi}_2 \in \mathbb{R}^{\sqrt{N} \times \sqrt{N}}$, so it takes $\sqrt{N} \sqrt{N} \sqrt{N}$ multiplications to compute $\tilde{\Phi}_1 X$, and an overall of $2N^{1.5}$ multiplications. A net saving of $\frac{1}{2} \sqrt{N}$ - if $\alpha = 4$ then in the 256 $\times$ 256 example above the separable scheme works 32 times faster. It also solves the storage problem.

The separable sampling scheme does allow efficient sampling of images compared to direct matrix-vector multiplication, but as not every sampling matrix is separable, the question of the performance loss due to the separable sampling scheme arise. Even if we do not limit ourselves to $\pm 1$ matrices, the question remains. i.i.d gaussian sampling matrices are considered good for CS, but using separable sampling with two i.i.d gaussian matrices is equivalent to sampling with the Kronecker product of the two matrices, which is no longer i.i.d or Gaussian. An interesting case is the \{0,1\} binary matrix - a Kronecker product of two binary matrices has large areas of zeros, which result in data loss and inability to reconstruct the image. The question of choosing matrices for separable sampling (both $\pm 1$ and general) is left for future work.

2.3.2 Separable Rademacher sampling matrix

We still need to show a way to use a separable sampling matrix with the MMA sampling device, in the special case when only binary \{0,1\} sampling is allowed. The Kronecker product of two binary matrices is problematic, but a \{-1,1\} matrix can be used instead in the following way: The Kronecker product of two \{-1,1\} matrices is also a \{-1,1\} matrix. First, we realize $\Phi_1, \Phi_2$ entries from the Rademacher distribution - $\pm 1$ w.p. 0.5 each. $\Phi$ is set by choosing $M$ random rows (the set $I_M$) from $\Phi_1 \otimes \Phi_2$. Denote $\Phi^B = 0.5 (\Phi + 1)$, were $1$ is a matrix or vector of ones, according to the context. $\Phi^B$ is a binary matrix, and thus can be used as the sampling matrix by the MMA. The MMA sample the image to obtain $y^B = \Phi^B x$. $y$ is obtained from $y^B$ in the following way: $y = 2y^B - 1x$. Note that $1x$ is just the summation of all the elements in $x$, so $y$ is obtained by using $M$ compressed samples and an additional
single sample that is the sum of \( x \)'s elements, which is obtained from the MMA. A similar solution has been independently developed by [17].

### 2.3.3 Examples

As in subsubsection 2.2.7, we now turn to test the CS scheme on images. First, we will test synthetic images - sparse in the spatial domain first, and then sparse under orthogonal and overcomplete transformations. In the second part of the experiments we will use real images.

There are two main differences between CS of 1D signals and of images. The first is the size of the signal, which is rather large in the case of images. The large signal size dictates the use of a separable sampling scheme (or any other scheme that does not involve direct \( \Phi x \) multiplication). The second difference is that roughly speaking, images are not very sparse, and does contain a significant amount of energy in the smaller coefficients of the chosen transform. While this may happen in any type of signal, in images it is almost always an inherent problem. Later on we will see how regularization techniques help in CS reconstruction under those conditions. The main source of reconstruction error is expected to be the difference between the sparse approximation and the image itself. In the synthetic 2D images we will add noise to \( x \) instead of \( \Phi x \) in order to simulate this difference. Noise in the acquisition process will be ignored for now.

\( \ell_1 \) vs. OMP, noise

The signal \( X \in \mathbb{R}^{64 \times 64} \) represents a small \( 64 \times 64 \) image. The notation \( x \) is reserved for the column stacking of \( X \). We now have that \( N = 4096 \) and again we use a \( 1/4 \) compression rate: \( M = 1024 \).

Sampling is done by calculating \( Y = \Phi^T X \Phi \), and taking \( M \) random elements from \( Y \) as \( y \). \( \Phi \) is a \( 64 \times 64 \) random \( \pm 1 \) matrix.

In this first experiments we change the sparsity of the images and the noise level. We use \( \ell_1 \) minimization and OMP.

Noise: the noise in the images case will not be added to \( \Phi x \), but will be the non-sparse part of the image. In the simulation, we take the zero coefficients and set them to random values realized from the Laplacian distribution, such that the energy of the noise will be a pre-determined fraction of the signal’s energy. Thus, we use SNR as the noise level parameter. The results of the first experiment are displayed in figure 7.

As in the 1D case, OMP performs better than \( \ell_1 \), although the run times are now comparable when \( ||x||_0 \approx 200 \). We learn from figure 7 that OMP will reconstruct a 4096 pixels image from 1024 measurements when it has 150 significant coefficients, and the energy in the remaining coefficients is 1% of the signal energy (20db).

CS of images with dictionaries

Images are rarely sparse in the spatial domain, so we should check how well can we reconstruct images that are sparse in different dictionaries. The treatment of orthogonal and overcomplete dictionaries is unified in this paragraph. There are many "specialized" transform for images, such as curvelets [36], contourlets [38], that represent images more sparsely. In this experiment we test orthogonal wavelets (db2) and the overcomplete dual-basis dictionary.
Figure 7: CS of synthetic sparse images, $\ell_1$ reconstruction (a) and OMP reconstruction (b). Each line stands for different noise level (SNR). The x-axis indicates the sparsity of the signals, and the y-axis indicates the probability of success in CS reconstruction. Success is defined as identifying the correct $s$ non-zero entries of the signal.
Figure 8: CS of synthetic sparse images with dictionaries, OMP reconstruction. Each line stands for different dictionary. The x-axis indicates the sparsity of the signals, and the y-axis indicates the probability of success in CS reconstruction. Success is defined as identifying the correct $s$ non-zero entries of the signal.

DCT+WT(db2). Noise level is 20db. The results are displayed in figure 8. In addition to figure 8, CS reconstruction was done on a $128 \times 128$ with the wavelets db2 dictionary. The results are displayed in figure 9 and will be used when we will come to deal with real world images in subsubsection 2.3.4.

The use of orthogonal bases decrease performance slightly, were the use of overcomplete dictionary oblige the image to be more sparse. However, we use overcomplete dictionaries for this reason exactly - to have very sparse representations. Run time considerations: the linear transformations add to the complexity of the reconstruction process, making it much slower. In the 1D case, the linear transformation was implemented as a matrix-vector multiplication, and embedded into the sampling matrix: $A = \Phi \Psi$, $y = Ax$. In the 2D case we do not have $\Phi$ explicitly. If the transformation could be implemented in a separable manner, we could have combined the transformation and the sampling. In the chosen transformations this is not the case - a single level of the wavelets decomposition is separable, but not all the levels together, for example. There is always the option to combine the first level decomposition with the sampling (and the adjoins), and then take the appropriate part of the image and continue with the transformation. We will not do this here to keep things simple.
Figure 9: CS of synthetic sparse 128 × 128 image, OMP reconstruction. Each line stands for different dictionary. The x-axis indicates the sparsity of the signals, and the y-axis indicates the probability of success in CS reconstruction. Success is defined as identifying the correct $s$ non-zero entries of the signal.

2.3.4 Real world images

The time has come to try compressive sampling on real-world images. As was already mentioned, images are usually not exactly sparse, but compressible - most of their energy is concentrated around small number of wavelets coefficients. In addition and contrary to the 2D experiments, the approximation coefficients tend to have more energy than the details coefficients. Also, images tend to have low total-variation [13]. Those properties will be used for the reconstruction and for developing of novel sampling scheme.

The images The chosen images 128 × 128 grey-scale images. The images are displayed in figure 10.

Sparsity of images According to figure 9, a 600-sparse 128 × 128 image can be reconstructed when the the energy of residual image ($I_r = I - I_a$) is about 1% of the energy of the image (SNR=20db). The sparsity of the four test images will be tested by calculating a 600-sparse approximation under different transforms, and calculating the SNR. Only critically sampled transforms were tested - different wavelets bases. The results are displayed in figure 11.

The conclusion from the graph is that we expect to be able to reconstruct the 600-sparse approximation of the "R", "moon" and the "peppers" images, and that the "cameraman" image reconstruction may not work very well. The appropriate wavelets transforms are: R image: db1 (Haar), moon: bi-orthogonal 4.4, peppers: bi-orthogonal 4.4, cameraman:
Figure 10: The images that will used in subsection 2.3.4.
bi-orthogonal 4.4.

**Performance evaluation** When it comes to images, Peak-Signal to
Noise Ratio, PSNR is an accepted measure of performance. The PSNR
of image $\hat{I}$ with regard to image $I$ is defined as:

$$\text{PSNR}(I, \hat{I}) = 10 \log_{10} \frac{\text{max} I^2}{\text{MSE}(I, \hat{I})}$$

(9)

were

$$\text{MSE}(I, \hat{I}) = \frac{1}{N} \sum_{i,j} (I_{i,j} - \hat{I}_{i,j})^2$$

(10)

It is also accepted to take $\text{max} I$ as 1 (or 255), regardless of the specific
image $I$.

**Trying compressive sensing** A compressed sampling will be done
on each image (1/4 compression), and the reconstruction results will be
displayed together in figures 12 - 15 with the original image and the best
sparse approximation. In addition, it is interesting to look at a much
simpler compression scheme: under-sample the image uniformly, and re-
store the image by interpolation (by using Matlab’s \texttt{imresize} function,
for example). There is no hope of getting perfect reconstruction, but it
will serve as a simple benchmark so the usefulness of compressive sensing
can be assessed.

Some details about the reconstruction process: OMP reconstruction was
used, although for relatively large signals such as $128 \times 128$ images, $\ell_1$ minimization is faster than OMP. The results are quite similar. The results are displayed in figures 12 - 15.

In first glance, the results are disappointing: CS fails to perfectly reconstruct the images, or at-least reconstruct the optimal sparse approximation. The R letter image (figure 12 is the only example where CS reconstruction is successful, and outperforms the naive uniform sub-sampling and interpolation in terms of SNR. The moon example result is also acceptable. Generally speaking, the uniform sub-sampling scheme result in pleasant looking but blurred images, while the CS reconstruction is not blurred, but suffers from many artifacts. There are two reasons for the reconstruction failures: either the images are not sparse enough under the orthogonal (or bi-orthogonal) transforms that were chosen, or the distribution of the significant coefficients is not random, as was the synthetic simulations. The question now is what can be done? Is there any hope

Figure 12: ‘R’ letter image (a), sub-sampling and interpolation (b), optimal sparse approximation (c), and CS reconstruction (d).
Figure 13: Moon image (a), sub-sampling and interpolation (b), optimal sparse approximation (c), and CS reconstruction (d).
Figure 14: Peppers image (a), sub-sampling and interpolation (b), optimal sparse approximation (c), and CS reconstruction (d).
Figure 15: Camera image (a), sub-sampling and interpolation (b), optimal sparse approximation (c), and CS reconstruction (d).
of applying compressive sampling to data that is not sparse enough? The answer is yes. For example, Total Variation was used as a regularization term by [18]. But before we discuss improving the reconstruction process, we are going to suggest a unique sampling scheme.

2.3.5 LP-HP compressive sensing

Many images are not sparse when using critically sampled transformations. Specifically, when it comes to wavelets, the approximation part of the multi-scale wavelets transform is usually dense - most of its coefficients are larger than zero. Of course, there are examples where large part of the image is dark, such as the "moon" or the "R" images above, but those are the cases were CS worked reasonably well in subsubsection 2.3.4. As for the other images, their approximation is quite full. Earlier in this report, it was mentioned that if we knew which coefficients are the largest (in absolute value), we would have sampled them directly. We do not know that, and therefore use random sampling matrices for compressed sensing. However, it seems that we do know, at least for many images, that the approximation coefficients are expected to be high. We can thus divide the sampling into to parts: sample the approximation coefficients directly, (Low-Pass, LP stage), and then sample "the rest of the image" in a compressed manner (High-Pass, HP stage). By "the rest of the image" we mean an image who’s approximation part has been removed - such an image is hopefully sparser and easier to reconstruct. To be consistent, we will limit the overall samples to $M$.

Some notations: denote $M_L$ as the number of low-pass samples, and $L_H$ as the number of regular CS samples. We set $M = M_L + M_H$. Let $x = \Psi z$, and let $z_L$ be a vector who’s first $M_L$ coefficients are the same as of $z$ - the approximation part, and the rest are zero. Let $z_H = z - z_L$ be the details coefficients. Let $x_L = \Psi z_L$, and $x_H = \Psi z_H$. We get that $x = x_L + x_H$. The sampling is done by two sampling matrices: $\Phi_L$ and $\Phi_H$. We have $y_L = \Phi_L x$, and $y_H = \Phi_H x_H$. $x_H$ is not given explicitly, so we take $\tilde{y} = \Phi_H x$, and $y_H = y - \Phi_H x_L$. Taking $\Phi_L$ to be the first $M_L$ rows of inverse $\Psi$ will result in $y_L$ equal to the first $M_L$ elements of $z$.

Thus, we have $z_L$ and therefor $x_L$ exactly, and $y_H$ can be obtained. We now solve the problem of reconstructing $z_H$ only, a problem that should be easier than reconstructing $z$, as $z_H$ is more sparse. The reconstruction error is due to the reconstruction of $z_H$ only, as we have $z_L$. The proportion $M_L/M$ should be chosen - the extreme case, when $M_L/M = 1$ is the naive approximation sampling, and $M_L/M = 0$ is ordinary compressive sensing. Not any number can be chosen for $M_L/M$ - it depends on the structure of the transform. In 2D wavelets decomposition, $M_L$ increases (or decreases) by a factor of 4.

LP-HP CS - experiments Let us try the new LP-HP CS scheme on the data from subsection 2.3.4. Let $M_L/M = 1/4$. The results are displayed in figure 16.

In the moon, peppers and cameraman images, the results are superior to those obtained by CS in subsection 2.3.4, especially the $\ell_1$ reconstruction. In the R-letter image case, the result of the OMP reconstruction
Figure 16: LP-HP compressive sensing. Left column - original 128 × 128 image. Middle column - OMP reconstruction. Right column - $\ell_1$ reconstruction.
is the same as in 2.3.4. It seems that when the image is "full", it is much better to use the LP-HP scheme. When we expect to have a mostly empty image, then we should use ordinary CS. Another incentive for using LP-HP approach is when the LP part is very easy to obtain - for example, when using compressive sensing for very fast analog-to-digital conversion [19], sampling the signal at a lower rate is usually much easier, and thus the CS part will be used for "fine-tuning" of the digital sampled signal. Another example which is going to be explored in this work, is the use of ordinary, of-the-shelf RGB camera as the LP component of a CS based hyper-spectral imaging system.

2.3.6 CS regularization

After the sampling is done, we are free to use any technique for reconstruction. As images tend to have low Total Variation, (TV), [13], we can add a TV term to the reconstruction process. Total-variation of an image is defined as:

$$TV(I) = \int \int \left\| \left( \frac{dI(x,y)}{dx}, \frac{dI(x,y)}{dy} \right) \right\|_2 dy dx$$  \hspace{1cm} (11)

in the continuum, and in the discrete case we use finite differences:

$$TV(I) = \sum_{i,j} \left\| (I_{i+1,j} - I_{i,j}), (I_{i,j+1} - I_{i,j}) \right\|_2$$  \hspace{1cm} (12)

The use of TV term is equivalent to demand sparsity of a finite difference transform. This may be accurate in case of piecewise constant function, but in the general case the TV term does not relate directly to the sparsity of the image (in case of images). In CS, many times the reconstruction errors are artifacts rather than blur, and the TV regularization scheme should be effective in this case. The most natural way to incorporate the TV term is to use the $\ell_1$ minimization scheme - minimize both the $\ell_1$ norm of the coefficients and the TV of the image, with the fidelity constraint: $\|Az - y\|_2 \leq \epsilon$. We will need to switch from the QCQP solver to general non-linear solver, though. Another way to incorporate a TV term in the CS reconstruction is by generalizing OMP - see subsection 2.4.

**CS with TV - optimization** The optimization problem now becomes:

$$\min_{z} \|z\|_1 + \omega TV(\Psi z)$$

$$s.t. \quad \|Az - y\|_2 \leq \epsilon$$  \hspace{1cm} (13)

We can no longer use the QCQP solver, and must consider other algorithms. [32] is an excellent survey of optimization methods for the $\ell_1$ regularization of general (twice differentiable) loss function. It also contains numerical experiments comparing different methods. Generally speaking, the most efficient methods for dealing with the non-differentiable $\ell_1$ norm are by using constraint optimization approach. In those methods, we replace the optimization variable $x$ by $x = x_+ - x_-$, and adding the constraints $x_+ \geq 0$, $x_- \geq 0$. The size of the new optimization problem is thus double than the original one, which will be highly undesirable when
we will use compressive sensing for hyperspectral data acquisition, due to the very high dimensionality. Another group of methods, termed "Subgradient" approaches may also have good performance when we have a good initial guess for the solution, in terms of its active set (non-zero coefficients). In CS we will usually not have such a guess if the problem is not very sparse (in which case we would guess the zero vector). The last approach is the differentiable approximations, which replaces the \( \ell_1 \) norm by smooth surrogate function. The smooth surrogate function method does not require good initial guess, and does not change the dimensionality of the problem. However it might lead to slow convergence when the surrogate function is not smooth enough. This problem is solved by carefully adjusting the surrogate function sharpness parameter during the optimization [32]. The slow convergence is caused because the hessian matrix of the smooth approximation becomes ill-conditioned as the accuracy of smooth approximation increases. Another way around that is presented in [31], where a method similar to augmented lagrangian [30] for unconstrained sum-max problem is developed, allowing accurate solution with smooth surrogate functions. [31] may be naturally extended to the case of constraint optimization with sum-max terms, which is exactly (13), with the \( \ell_1 \) term being a special case of a sum-max term (see [31] for details). Following [28], the surrogate function for \( |z| \) we will use are:

\[
h_{\epsilon}(z) = |z| - \epsilon \log \left( 1 + \frac{|z|}{\epsilon} \right), \quad \epsilon > 0
\]

and its derivatives:

\[
h_{\epsilon}'(z) = \frac{|z| \text{sign}(z)}{\epsilon + |z|}, \quad h_{\epsilon}''(z) = \frac{\epsilon}{(\epsilon + |z|)^2}
\]

(13) will be replaced by:

\[
\min_z \sum_i h_{\epsilon}(z_i) + \omega TV(\Psi z)
\]

s.t.

\[
\|A z - y\|_2 \leq \epsilon
\]

(16) will be solved by the augmented Lagrangian method. The inner optimization problem will be solved by \texttt{minFunc} [33], which uses L-BFGS in this case.

**CS with TV - results** The results of the regularized CS are displayed in figure 17. The results are the best so far, better than the LP-HP CS. One should note, however, that the TV prior is not suitable for all images. Also, OMP and other established greedy methods cannot be used with the TV regularization.

2.3.7 LP-HP CS with TV

The next and final step (for images) is to combine the LP-HP sampling scheme and the Total-Variation regularization. The results are displayed in figure 18.

There is a significant improvement in comparison to the LP-HP only, and smaller improvement (~ 1 db) in comparison to TV regularization.
Figure 17: TV-regularized compressive sensing. Left column - original 128 × 128 image. Right column - $\ell_1 + TV$. 
Figure 18: LP-HP + TV-regularized compressive sensing. Left column - original 128 × 128 image. Right column - LP-HP + TV ($\ell_1$) reconstruction.
only for 3 out of four of the images. In each one of the four example images, the LP-HP compressive sensing combined with total variation give good results, both visually and in comparison to uniform sub-sampling - see figure 19.

2.3.8 CS of images - conclusion

A concluding graph of the different reconstruction methods on the different test images is displayed in figure 19. Throughout the last subsection we tested the CS scheme on synthetic and real-world images. We saw that while real-world images are often not sparse enough to allow the use of compressive sensing, by using total-variation regularization and a novel LP-HP CS scheme, we were able to achieve acceptable reconstruction performance in all four examples. Moreover, in addition to the PSNR performance measure, the CS reconstruction is less blurry than the simple sub-sampling one. One should note that we used a separable sampling scheme with a ±1 sampling matrix. More general sampling schemes may give even better results. We used that sampling scheme because it suits the physical sampling scheme that will be used in section 3.

2.4 Generalized OMP - GOMP

In the previous subsection, total-variation regularization was found to be crucial to the success of CS of images. Alas, TV regularization does not integrate into the OMP algorithm - in OMP we do only quadratic minimization, and TV is not a quadratic function. Other greedy algorithms, such as ROMP [12] and CoSaMP [?] does not incorporate non quadratic functions as well. We are forced to use the ℓ₁ minimization framework. In this subsection we develop the Generalized OMP - OMP with non-linear optimization process inside it, allowing the use of TV regularization in compressed sensing, for example. In OMP there are two minimization stages: the first is minimizing the residue over the already selected atoms. The second is finding the next atom that matches best with the residue. In GOMP, the first minimization will be general non-linear minimization, allowing the incorporation of TV, for example. The ‘matching’ part can be done either by simple scalar product as in OMP, or by minimizing a more complex problem that include the non-linear elements, were the minimization is done along a single atom axis each time - similar to Parallel Coordinate Descent (PCD). Further investigation into this algorithm is left for future work.
Figure 19: CS of images performance (PSNR) summary - OMP/$\ell_1$ reconstruction, LP-HP CS, TV regularization and LP-HP CS combined with TV regularization. The rightmost bar in every bar group is the simple 1/4 sub-sampling and interpolation performance.
3 Compressed sensing of hyperspectral data

3.1 Hyperspectral imaging

3.1.1 Background and notations

Hyperspectral (HS) imaging refers to imaging the electromagnetic (EM) waves reflectance/emitance/transmission properties of a scene or an object. A regular camera does just that over a specific range of the EM spectrum, with low spectral resolution - the famous RGB color base. In HS imaging, the spectral range may be wider (ultra-violet to deep IR), and the spectral resolution is much higher - from one hundred to over two thousands spectral bands. Thus, while the output of a regular camera is a three \( N \times N \times N \) matrices, a matrix for each of the R,G,B bands, the output of an HS imaging system is an \( N_s \times N_y \times N \lambda \) data cube. There are \( N \lambda \) spectral bands. RGB basis is very useful for visualization, as it matches the human visual system, so it can be used to capture and display a scene appearance such that it will look similar to the real scene in the eyes of a human observer. There is, however, literally 'more to the picture than meets the eye' - inspection of a narrow band of the EM spectrum may reveal additional information on a scene. For example, synthetic colors have very different spectral signature compared to natural colors. This fact is useful when searching for disguised men-made objects - an RGB aerial image may fail to recognize a green disguise-net as such because it does look like vegetation, while HS imaging system will easily discern the artificial object due to its different spectral signature. See [43] for an algorithm for anomaly detection in HS aerials. Another uses include inspection of forest health, and uses as a diagnostic tool in medicine are being researched.

3.1.2 Acquisition methods

Imaging a grey-scale image is done by projecting the light coming from a scene onto a an imager, which is usually a flat chip that detects EM energy with spatial resolution. The imager output is a 2D array of numbers. In color imaging we do the same, only this time the imager pixels are covered with red, green and blue filters, in some layout. Each pixel "sees" only one color, and we use spatial interpolation to obtain the R, G, and B plane. In acquiring hyperspectral data cubes, however, we obtain a three-dimensional cube, and cannot acquire it in a single shot. There are three approaches for HS imaging (below), but all of them are based on sequentially sensing 2D images, and composing them into a single data cube. In this work we will focus on the diffraction method, but the other approaches are also described for completeness.

Filter-based methods This is the most intuitive method: sequentially imaging a scene through a changing narrow-band color filter. In each shot we obtain a single spectral band of the HS cube. One major
drawback of this method is that we lose photons in the process - the filters work by swallowing all photons except the ones with a specific wavelength.

**Interferometry** An interferometer is a device that measures the interference pattern of a ray of light by splitting the light into two beams and creating a phase difference between them. The beams travel slightly different distance and then combined, creating interference. By gradually changing the distance the beams travel and taking measurements, the inverse fourier transform of the spectrum is obtained. One advantage of interferometry is the high spectral resolution that can be obtained.

**Diffraction** In diffraction based methods, the light from a single row in the scene is projected on a prism or diffracting grid and scattered to its different spectral components. The scattered light is projected on an imager, creating a 2D array whose one dimension is spatial, and the other spectral (the $\lambda$ axis). The scene is scanned line by line (for example by a rotating mirror), creating the HS data cube. This method is especially suitable for acquiring aerials from airplanes and satellites - the scanning is done naturally by the progress of the vessel.

In all three methods of HS data acquisition there is a trade-off between acquisition light and SNR: the faster we acquire each band (or line), the less photons we get, decreasing SNR. The same problem exist in color imaging as well, but in HS imaging we have much lower energy per band, and many more bands. The slow acquisition process is problematic when we try to image a changing scene. Another difficulty is the huge amount of data that needs to be stored or transmitted. In this work we propose an HS sensing device based on compressed sensing. The compressed sampling is done using unique hardware, and the reconstruction process combines the $\ell_1$ methods with novel techniques.

### 3.2 A Compressed sensing based system for HS data acquisition

#### 3.2.1 Sensing mechanism

We use the diffraction method (subsubsection 3.1.2) combined with the Micro-Mirror Array (MMA) (see subsection 2.3) to enable compressed hyperspectral sensing. The system is based on the diffraction acquisition mechanism, but the projection of an image "line" is done by the MMA rather than by a moving mirror. The structure of the system is as follows: an optical system focuses the image on the MMA. The MMA mirrors are placed such that in the "on" position, the rays coming from the lens are reflected towards another optical system, which concentrates the incoming rays on a prism (or diffracting grid). The prism and the second optical system is laid such that the horizontal resolution ($x$-axis) is maintained, but the vertical axis can be sampled one point at a time. The "line" that is projected by the MMA on the prism is scattered to its spectral components, and the spectral image of the line is recorded by an imager. The system is portrayed in figure ??.
general sampling schemes than scanning each line at a time - we can scan more complex patterns, and combine different pixels within each column. This way we can have a practical sampling schemes for compressive sensing - by using linear combinations of pixels with arbitrary vertical location \((y)\). The combination may be different for every column. This way we can have a sampling matrix that "mixes" the signal properly. The sampling matrix is still limited to mix only inside the columns\(^1\) - we will see that this has negative impact on the ability to correctly reconstruct the HS cube. Compressive sensing will be done by projecting randomly chosen pixels from each column on the prism, and repeating the process for \(N_y/\alpha\) times, were \(N_y\) is the number of pixels in the vertical \((y)\) dimension, and \(\alpha\) is the compression ratio. We sample \(1/\alpha\) of the samples we usually would in the direct sampling process. Hopefully, by using clever reconstruction schemes we will be able to reconstruct the image from the under-sampled data. There is a unique sampling matrix for each column \(x - \Phi_x\). Due to the structure of the optical system, it is not possible to use the separable sampling from subsubsection 2.3.1. Having one sampling matrix for each point on the \(x\)-axis, and having the same sampling matrices in all the bands is equivalent to using sparse binary sampling images [16] that repeat themselves. The complexity and storage issues are thus resolved, but not without cost - see the end of subsubsection 3.3.4. Mathematically, we have:

\[
y^{x,\lambda} = \Phi_x F^{x,\lambda}
\]

were \(F\) is the data cube, and \(F^{x,\lambda}\) is the \(x\) column in the \(\lambda\) band, and \(y\) is obtained by concatenating the \(y^{x,\lambda}\) vectors for all \(x, \lambda\).

### 3.2.2 RGB image

In addition to the mixed hyperspectral data, in parallel to the main optical system there is a standard RGB imaging system. As we will see, the RGB (or RGB-IR) image contribute to the accuracy of the reconstruction. This is due to the fact that we loose the spatial structure in compressive sensing, and try to restore it in the reconstruction process. The RGB image retains the spatial structure, but has low spectral resolution. It is equivalent to the LP-HP CS approach from subsubsection 2.3.5.

### 3.2.3 Noise

The noise in practice is dominated by shot-noise, in addition to dark noise and quantization noise. We will simulate a simple white Gaussian noise (the dark-noise). Noise is added both to the compressed sampling vector \(y\), and to the RGB-IR image \(I_{RGB}\). The noise levels are 40 db SNR for \(y\), and 60 db SNR for \(I_{RGB}\).

---

\(^1\)Either columns or rows - the \(x\) and \(y\) axes can be switched in the discussion. Moreover, one can imagine a system which does some of the mixing in the \(y\) dimension, and then in the \(x\) dimension. Such system will use additional optics.
Figure 20: Hyperspectral imaging CS-based acquisition system.
3.3 CS of HS data - Reconstruction

3.3.1 Overview

In this subsection we will develop the reconstruction algorithm of the compressively sensed HS data step by step. First, we will introduce the data cube that we will work on and analyze its sparsity under different dictionaries. Second, we will apply reconstruction algorithms to the sampled data, in each algorithm we will add terms and constraints that will increase the accuracy of the reconstruction. The reconstruction algorithms we will use are:

- Simple $\ell_1$ algorithm (plain CS)
- CS + Total Variation term
- LPHP CS + TV - RGB image
- LPHP CS + TV - column means

At the end of this section we will compare the results, analyze the sources of error and suggest further methods to improve the reconstruction.

3.3.2 The data

Unlike color images, which are readily available, hyperspectral data is harder to come by. AVIRIS [42] project is an important source, but it seems that the aerals from AVIRIS tend to have simple spectral behavior, and were "too easy" to reconstruct - by interpolating the RGB image alone one could have a reasonable reconstruction. This is not good news though - if there is something small with different spectral signature, it will be lost in such simplified approach. We use the data obtained from Surface Optics Company (SOC), which kindly allowed me to use two of their HS data cubes. We will mainly use small part of the ‘Leaf’ data cube that contains a variety of materials in the scene - leafs, wall, a plastic cup with some printing and the SOC HS camera. See figure 21. The data cube size is $64 \times 64 \times 120$ (120 spectral bands), with spectral range of 412 - 908 nm. Close look on the individual bands of the 'Leaf' data cube reveals that they are rather noisy. For the experiments ahead it is preferred to have "clean" data. Therefor Total-Variation De-Noising was applied to the data by solving: $x^{TV} = \text{argmin}_x \{TV_x(x) + \omega \| x - \hat{x} \|_2^2 \}$, and chose $\omega = 0.33$ by visual inspection. The overall PSNR (with the noise defined as $x - x^{TV}$) of the chosen smoothed HS cube is 54 db. The $TV_x$ operator is the sum of the 2D total-variation of all the bands.

3.3.3 Dictionaries and sparsity

As sparsity is a key issue in compressive sensing, we should find a dictionary in which the HS cube is sparse. The first natural choice is the 3 dimensional separable wavelets transform. The 3D separable WT is done by applying a 1D WT on each dimension, obtaining eight bands: Approximation, and seven details bands. The transform is then applied to the approximation, and so on.
The spectral axis, $\lambda$, tends to be highly correlated for each spatial location. Also, many adjacent spatial location have a very similar spectral signature. Those two observation give us hope that the HS data cube will have a sparse representation, compared to the 2D case. Figure 22 display the sparsity of the TV de-noised ‘Leaf’ data cube ($x^{TV}$) under 2D WT (only spatial), 1D WT (spectral axis) and the 3D separable WT - the last result is much better.

The bi-orthogonal wavelets were chosen because they tend to have good performance for sparse representation. Exhaustive search over all the WT types was not done, because in real life we will not have the original data to allow us to choose the best transform. We will need to use good general transforms. (It is possible, though, to do several reconstruction using different dictionaries until the result is satisfactory).

Observing figure 22, we come to the conclusion that the $\lambda$ axis is not well represented by the wavelets basis, and it might be better to use overcomplete dictionary in the $\lambda$ axis in the reconstruction part.

One may also consider using non-separable overcomplete 3D transforms - 3D versions of Dual-tree wavelets [35], Curvelets [37] or Contourlets [38]. The complexity of calculating those non-separable transforms is much larger than that of separable transforms, which renders the reconstruction process infeasible on the hardware we used. However, by using different hardware and optimizing the code it may be possible to use those transforms. Furthermore, the SESOP [28] algorithm deals with the case of efficiently optimizing objective functions of the form $f(Ax) + g(x)$, were $A$ is a linear transformation that is computationally demanding compared
Figure 22: Sparsity of the ‘Leaf’ data cube in different transformations - 2D (spatial domain, blue), 1D (spectral axis, green), and 3D (red). The sparsity of the HS cube itself is also presented (none).
to \( f(\cdot) \) and \( g(\cdot) \).

**Overcomplete 2D dictionaries** Overcomplete transformations, such as curvelets [36] and contourlets [38] have been known to improve images sparsity. We will not use them for now, though, because of the computational complexity.

### 3.3.4 Reconstruction - plain CS

Now that we have a sparsifying base to work with, we will try to reconstruct the sampled HS cube using the \( \ell_1 \) scheme:

\[
\min \{ \| z \|_1 \text{ s.t. } \| Az - y \|_2 \leq \epsilon \} \tag{18}
\]

were \( A = \Phi \Psi \), \( \Psi \) is the inverse of the sparsifying transform, and \( \Phi \) is the sampling operator. Here we will use an unconstraint optimization, similar to BPDN (Basis Pursuit De-Noising):

\[
\min \{ \| z \|_1 + \omega \| Az - y \|_2^2 \} \tag{19}
\]

By tuning \( \omega \), we can obtain results that are equivalent to the constraint setting. Note that (19) is the Lagrangian of (18), and thus by searching over \( \omega \) we get the solution to (18).

**Reconstruction results evaluation** There is a question of how to present and evaluate the results? While in 1D and 2D we can simply plot a graph or an image, the HS data cube is composed of many (120 in our case) images, or alternatively 64 \( \times \) 64 signals. In order to compare reconstruction performance, we will use three different views on the error:

The first is to display the overall PSNR, the second is a graph of the PSNR in each band, and the third is an image of the spectral axis SNR in each spatial pixel. In addition, selected bands from the original HS cube and from the reconstruction will be displayed. Regarding PSNR: the peak signal value is taken to be the maximum element of the cube \( X \), \( X_{\text{max}} \), and not the upper limit of the dynamic range - \( 2^{16} - 1 \) because the HS acquisition device doesn’t use the entire dynamic range. Let \( X_{i,j,k} \) be the pixel at the spatial location \( i, j \) and in the \( k \)'th spectral band. The MSE between data cubes \( X \) and \( \hat{X} \) is defined as:

\[
\text{MSE}(X, \hat{X}) = \frac{1}{N} \sum_{i,j,k} (X_{i,j,k} - \hat{X}_{i,j,k})^2 \tag{20}
\]

Were \( N \) is the number of elements in \( X \). The PSNR between two HS cubes is defined as:

\[
\text{PSNR}(X, \hat{X}) = 10 \log_{10} \frac{X^2_{\text{max}}}{\text{MSE}(X, \hat{X})} \tag{21}
\]

When evaluating the PSNR of a single band, \( X^k \), we will define \( \text{PSNR}_1 \), which is the PSNR calculated with the single cube maximum, and \( \text{PSNR}_2 \),
which is calculated with the specific band’s maximum. Let $X_{k}^{\text{max}}$ be the $k$'th band maximal value. The MSE between the $k$'th bands of $X$ and $\hat{X}$:

$$\text{MSE}(X_{k}, \hat{X}_{k}) = \frac{1}{N_{xy}} \sum_{i,j} \left( X_{k}^{i,j} - \hat{X}_{k}^{i,j} \right)^{2} \tag{22}$$

$$\text{Were } N_{xy} \text{ is the number of spatial pixels in } X_{k}. \text{ PSNR1 and PSNR2 are defined as:}$$

$$\text{PSNR1}(X, \hat{X}) = 10 \log_{10} \frac{X_{\text{max}}^{2}}{\text{MSE}(X_{k}, \hat{X}_{k})} \tag{23}$$

$$\text{PSNR2}(X, \hat{X}) = 10 \log_{10} \frac{(X_{\text{max}}^{k})^{2}}{\text{MSE}(X_{k}, \hat{X}_{k})} \tag{24}$$

The separation to PSNR1 and PSNR2 is important when looking at bands with low intensity (UV and IR bands), where the PSNR1 level may be high, but the PSNR2 level is low, indicating that after proper illumination scaling, we will have a noisy image.

**Optimization and convergence**

As mentioned above, we will use unconstrained optimization with weight $\omega$ on the sampling term as a substitute to constraint optimization (18). There exist a specific value for $\omega$ such that the solution for the unconstraint problem (19) will be equal to the solution of (18), as (19) is the Lagrangian of the constraint optimization problem (18), with $\omega$ the lagrange multiplier (we discard $\epsilon$ as it does not affect the minimal point). Solving (19) is a matter of tuning $\omega$ such that the sampling term will have the desired value at the minimum, by bisection for example. Using unconstraint setting was found to be much faster than solving (18) by the penalty method, probably because the quadratic constraint function becomes a power of 4 function after applying the quadratic penalty function. The minimum of (18) is found, but that doesn’t guarantee a perfect (or even reasonable) reconstruction, as the original signal itself does not minimize the optimization problem. We can say that in a way, the goal in compressed sensing is finding an optimization problem who’s minima is the desired signal. Hence, my target in solving CS problem by convex optimization is to reach a signal all of who’s terms ($\ell_{1}$, total-variation and sampling for example) are smaller or equal to those of the original signal. When such a signal is reached, there is very little hope that further optimization will improve the reconstruction. If the result is not good enough, then the optimization problem should be refined.

**Plain CS - results**

PSNR1 (23) and PSNR2 (24) in each band are displayed in figure 23, and the overall PSNR is 27.1 db. An example band is displayed in figure 24.

The results are far from satisfactory. This might be a surprise, given that in figure 22, for a 4% sparse approximation we have 40db PSNR, meaning that the HS data cube is rather compressible. The problem is that $\Phi$, the sampling matrix, is not a simple random matrix - it is mostly empty, and the mixing is done only within the columns. Moreover, for fixed $x$, the mixing is the same for all the bands. It is interesting to check
Figure 23: PSNR per band of simple ($\ell_1$) CS reconstruction. PSNR1 (23) in blue, PSNR2 (24) in red. Overall PSNR is 27.14 db.

Figure 24: Comparison between the original HS cube and the $\ell_1$ based reconstruction of band #40, 21.6 db PSNR. very poor performance.
what will be the performance if we use the separable 2D sampling scheme, with different sampling matrices for the different bands. The results are displayed in figures 25 and 26 with 31.8 db PSNR. As expected, there is a significant improvement over the results of the columns-only mixing. They might get even better if the mixing will be done between bands as well. In any case, we have to stay with the column (or rows, see footnote in page 38) mixing and identical Φ matrices for all the bands, because this is the sampling architecture dictated by the physical system.

### 3.3.5 Reconstruction - CS + \( TV_{xy} \)

We have seen in subsubsection 2.3.4 that total-variation regularization improve the reconstruction. We add the \( TV_{xy} \) term to the objective function, were \( TV_{xy} \) is the sum of the TV of each band separately: \( TV_{xy} = \sum_{\lambda} TV_{2D}(X^\lambda) \), were \( TV_{2D}(\cdot) \) is the TV of image operator, and \( X^\lambda \) is the image in the \( \lambda \) band. The optimization problem will now be:

\[
\min_{z} \left\{ \|z\|_1 + \omega_1 TV_{xy}(\Psi z) + \omega_2 \|Az - y\|^2 \right\}
\]  

#### Optimization and convergence

While in (19) we had a single scalar, \( \omega \), to tune, in (25) we have two, which makes the tuning of the \( \omega \)'s much harder. Furthermore, we will have more terms next - for the RGB image, for example (see below). In order to avoid the need for tedious search over the \( \omega \)'s, we use the Soft Shrinkage operator [26] as a middle way between the constraint and unconstraint optimization. The idea is that the penalty function, \( \phi_\lambda(t) \), equals the (shifted) identity function for
values higher than \( \delta \), and zero otherwise:

\[
\phi_\lambda(t) = \begin{cases} 
  t - \lambda, & t \geq \lambda \\
  0, & t < \lambda 
\end{cases} \tag{26}
\]

\( \phi_\lambda(t) \) in (26) is not differentiable in \( \delta \), so a smoothed version must be used for optimization. Following [28] we use:

\[
\phi_{\lambda,s}(t) = \frac{1}{2} \left( t - \lambda - s + \sqrt{(\lambda + s - t)^2 + 4st} \right) \tag{27}
\]

See plots of \( \phi_{\lambda,s}(t) \) with different values of \( s \) in figure 27. In addition, all the terms and the constraints functions are normalized by their expected target values. For the sampling term (constraint), the target value is the known noise level. The normalization constants will be denoted by \( \lambda_{\text{term}} \), and the \( \lambda \) in \( \phi_{\lambda,s}(t) \) (27) will be set to 1, because we want the normalized constraints to be bellow 1 at the optimal solution. (25) will be replaced by:

\[
\min_z \{ \lambda_1 \|z\|_1 + \omega \lambda_{TV} TV_{\psi}(\Psi z) + \phi_{1,s}(\lambda_{\text{amp}}\|Az - y\|_2^2) \} \tag{28}
\]

Finding the correct value of \( \lambda_{\ell_1} \) and \( \lambda_{TV} \) is not straight-forward - the total-variation of the bands can be estimated from the RGB image (see below), and the \( \ell_1 \) norm of the HS cube can be estimated based on past experiments. Refinement of the relation between \( \lambda_{\ell_1} \) and \( \lambda_{TV} \) is done by adjusting \( \omega \) after initial optimization. Note that the only parameter that should be calibrated is \( \omega \), as the \( \lambda \)'s are pre-determined and fixed. Let
Figure 27: Smooth approximations of Soft Shrinkage: $\phi_{1,s}(t)$ in a., and the derivatives on b.
us write the objective function in (28) again in a symbolic form, omitting
the normalization constants:

\[ f(z) = f_{\ell_1}(z) + f_{TV_{xy}}(z) + \phi_{1,s}(f_{\text{samp}}(z)) \]  

(29)

Now we look at the gradient of (29):

\[ \nabla f(z) = \nabla f_{\ell_1}(z) + \nabla f_{TV_{xy}}(z) + \phi'_{1,s}(f_{\text{samp}}(z)) \nabla f_{\text{samp}}(z) \]  

(30)

As in the case of penalty method [40], (30) is the gradient of the augmented
lagrangian of:

\[ \min_s f_{\ell_1}(z) + f_{TV_{xy}}(z) \]

s.t. \( f_{\text{samp}}(z) \leq 1 \)  

(31)

As long as the lagrange multiplier of \( f_{\text{samp}}(z) \leq 1 \) is smaller than
1, \( \phi'_{1,s}(f_{\text{samp}}(z)) \) in the solution will be the lagrange multiplier. Since
we used a smooth \( \phi_{1,s} \), we don’t get the solution to (31) exactly - the
constraint will be \( f_{\text{samp}}(z) \leq \epsilon \), with \( \epsilon \) ”around” 1, depending on the
smoothness parameter \( s \). If we want a more accurate solution, we can
decrease \( s \), were in the limit \( s \rightarrow 0 \) we get the exact solution. However,
it is enough to use \( s = 0.05 \) - we get \( 0.5 < \epsilon < 1.5 \), which is enough here.
Increasing \( s \) makes the optimization problem much harder. The problem
converge to values lower than those of the original signal in each term
after less than 500 functions evaluations.

CS + TV_{xy} - results The results are displayed in figures 28 and 29.
We can see an improvement over the \( \ell_1 \)-only reconstruction, although
there is still place for improvement.

Figure 28: PSNR per band of \( \ell_1 + TV_{xy} \) CS reconstruction. PSNR1 (23) in
blue, PSNR2 (24)in red. Overall PSNR is 35.5 db.
Figure 29: Comparison between the original HS cube and the $\ell_1 + TV_{xy}$ reconstruction of band #40, 28.9 dB PSNR.

### 3.3.6 $TV_{\lambda}$

Adding a TV term in the $\lambda$ axis may contribute to the reconstruction, but the negative influence of total-variation regularization on the peaks in the spectrums should be taken into consideration. It may be possible to use a variation of the TV regularization that preserves peaks in the signal by using $\ell_p$ norm, with $p < 1$. This is left for future work.

**CS + TV_{xy} + TV_{\lambda} - results** The results are displayed in figures 30 and 31. There is very small improvement over the CS + TV_{xy} results. The reason is that the total-variation along the $\lambda$ axis of the CS + TV_{xy} reconstruction is already low compared to the original signal's $TV_{\lambda}$.

### 3.3.7 LPHP CS for HS

As in the 2D case, we would try to implement the principle of LPHP CS. The mirror-array system impose limitations on the ability to sample approximations of the HS data cube. In the spatial plane, the column (or rows)-only mixing allow only approximation in on dimension. The bands can be sub-sampled only in one direction each time, which is much less efficient. Because of the relatively small spatial resolution in our experiments, we will use limited LPHP approach by only sample the column mean.

LPHP CS will also be implemented in the spectral axis indirectly - by using the RGB image, which can be viewed as a low spectral-resolution approximation of the HS cube. See subsubsections 3.3.8 and 3.3.9.
Figure 30: PSNR per band of $\ell_1 + TV_{xy} + TV_{z}$ CS reconstruction. PSNR1 (23) in blue, PSNR2 (24) in red. Overall PSNR is 35.6 db.

Figure 31: Comparison between the original HS cube and the $\ell_1 + TV_{xy} + TV_{z}$ reconstruction of band #40, 28.9 db PSNR.
3.3.8 LPHP - RGB image

The first application of the LPHP principle will be by using the RGB image apparatus: in addition to the MMA part, there is an ordinary RGB camera parallel to the optical axis (see subsection 3.2.2). The RGB image (or, preferably RGB-IR) camera give us a low-spectral resolution (but full spatial resolution) approximation of the HS cube. This is somewhat different than the LPHP CS approach presented in 2.3.5, as the HP part is done along the y-axis only. Another view of the RGB image is that the MMA-based sampling of the HS data cube loses the spatial structure of the image, while preserving the spectral data. The RGB image preserve the spatial structure, but loses the spectral resolution. The reason behind this specific approximation is that RGB, and even RGB-IR images are readily available, small and cheap. Some details: The RGB-IR camera output is four $N_x \times N_y$ matrices, each for every R, G, B and IR bands.

The color filters on the imager are positioned according to the Bayer layout:

\[
\begin{array}{cccc}
R & G & \text{IR} & B \\
\end{array}
\]

While RGB cameras often use interpolation in order to achieve maximal resolution, here we can assume that an $128 \times 128$ imager is used and no interpolation is done in the demosaic process - every square of R,G,B,IR pixels is saved as a single 4 color pixel.

In order to use the data further, we need to know the transformation between the spectral signature and the response of the different color pixels. Such data is part of any imager specification. For simplicity we assume gaussian distributions around the R,G,B and near-IR parts of the spectrum. See figure 32. Let $\lambda^c$ be a vector of the HS cube central
wavelength in each band, and \( \lambda^w \) the "width" of each band. (the bands central wavelength may be not evenly distributed). Let \( \tilde{w}^R(i) \) be a vector of mean response of the red pixels around the corresponding wavelength: \( \tilde{w}^R(i) \) is the mean of the red pixels response to a \( \lambda^w(i) \) wide range of wavelengths around \( \lambda^c(i) \). Define the vector \( w^R \) by:

\[
w^R(i) = \frac{\lambda^w(i) \tilde{w}^R(i)}{\sum_i \lambda^w(i)}
\]

(32)

The red plane of the image will be thus calculated by:

\[
I^R(X)(x, y) = \sum_i w^R(i) X(x, y, i)
\]

The red plane of the image will be thus calculated by:

\[
I^R(X)(x, y) = \sum_i w^R(i) X(x, y, i)
\]

I^G(X)(x, y) = \sum_i \tilde{w}^G(i) X(x, y, i)
I^B(X)(x, y) = \sum_i \tilde{w}^B(i) X(x, y, i)
I^IR(X)(x, y) = \sum_i \tilde{w}^IR(i) X(x, y, i)

(33)

Denote the colors set as \( \text{color} = \{R, G, B, IR\} \). The RGB term (or constraint) will be:

\[
f_{RGB}(z) = \sum_k \| I^{\text{color}_k}(z) - I^{\text{color}_k} \|_2^2
\]

(34)

Were \( I^{\text{color}_k} \) without an argument is the sampled color plane, \( I^{\text{color}_k}(X) \) is a color plane calculated from the HS data cube \( X \), and \( I^{\text{color}_k}(z) \) is the color plane calculated from the \( z \) coefficients vector by first transforming it into \( X \). The optimization problem will be:

\[
f(z) = f_{\ell_1}(z) + f_{\ell_1\phi_1}(f_{\text{sample}}(z) + \phi_1, \phi_2, f_{\text{RGB}}(z))
\]

(35)

Again, \( f_{\text{RGB}}(z) \) will be normalized by its expected value - the noise level.

**CS + TV_{xy} + RGB image - results** The results of the RGB LPHP CS (plus TV_{xy}) are displayed in figures 33 and 34. Again, we can see an improvement - the PSNR2 in all the bands is above 30 db, and in most bands is above 35 db. The overall PSNR is 40.7 db. There is a new phenomenon of fluctuating performance between the bands - from 43 db PSNR2 to 31 db. The fluctuations seems to be related to the spectral positions of the the RGB-IR sensors. Later on, we will try to mitigate this phenomenon by tighter control over the errors in different bands separately.

### 3.3.9 LPHP - Column means

As the sampling of the approximation part of the wavelets transform by the mirror-array is infeasible in the current configuration, we will only sample columns means as a limited version of the LPHP CS in subsub-section 2.3.5. First, one of the rows (the first) in each of the sampling matrices is changed to be all ones. Thus, we have the column means. We can use them in several ways: firstly, just change the sampling and trust the sampling term to make the columns means close enough to their
Figure 33: PSNR per band of $\ell_1 + TV_{xy} + I_{RGB}$ CS reconstruction. PSNR1 (23) in blue, PSNR2 (24) in red. Overall PSNR is 40.7 db.

Figure 34: Comparison between the original HS cube and the $\ell_1 + TV_{xy} + I_{RGB}$ reconstruction of band #40, 36.8 db PSNR.
measured values. The second approach is to address the columns means specifically, forcing their error to be small regardless of the overall sampling error. The first approach was used, and the error of the column means was small enough, so there was no need for refinement. The reconstruction results, however, did not improve and were slightly worse (by 0.2 db). This doesn’t mean that the columns means may not improve the performance on different data, and we should bear in mind that for larger data cubes we are likely to use a more detailed LP approximation.

3.3.10 Non-negativity

Non-negativity is an accepted and straightforward constraint in image restoration problems, as the result image should be non-negative. This also true for HS cubes, and a non-negativity term can be added to the objective function. However, the results we have seen so far are non-negative anyway, so adding such a constraint will only make the optimization process longer (every objective function evaluation will take longer). If there will be negative values in future results, a non-negativity term will be added. We will use the following penalty for negativity:

\[ f_{\text{non-neg}}(x) = \| x^- \|^2_2, \quad x^-(i) = \begin{cases} 0, & x(i) \geq 0 \\ x(i)^2, & x(i) < 0 \end{cases} \]  

3.4 Summary

We have seen that HS data can be reconstructed from a four-times under-sampling, with reasonable performance. Total-variation regularization in the image plane have played a major role in allowing the reconstruction, as well as low spectral resolution approximation by an RGB-IR camera. The PSNR2 of the bands of the three reconstruction is displayed in figure 35. Still, there is place for improvement - we would like to have all the bands near the 40 db PSNR2 or above. The future work subsubsection below will outline our research plans in the near future.

3.4.1 Future work

In order to improve the compressed sensing of HS data reconstruction results, we will use the following novel approaches:

1) **Tight error control**  While the use of "overall error" terms (such as \( \| Ax - y \|^2_2 \)) is common in image restoration and de-noising, in the compressed sensing setting the error often tends to concentrates in specific areas, degrading the reconstruction appearance and PSNR. This is apparent in figure 35, were some bands have low PSNR2, while others has high PSNR2. In the future we intend to try and make the errors more uniform.

2) **Spectral axis sparsity**  Compressed sensing is based on the sparsity of the signal (although not entirely). Looking at figure 22, we see that critically-sampled 1D wavelets (bior4.4) transform of the spectral axis does
not really result in a sparse representation. The 3D transform is better, naturally, but finding a 1D transform that will give us sparser representation of the spectral axis is bound to improve the sparsity of the 3D transform. We will focus on the spectral axis for two reasons: first, inspecting the spectral signals reveals that they are rather simple, and similar to each other, which also invite us to use bootstrap dictionary-learning. Second, the linear transformations are implemented by matrix-vector multiplications and thus the complexity of the transformation increases only linearly with size of the dictionary - as long as it is either a 1D transformation (for the spectral axis), or a separable 2D transformation.

3) Spectral axis dictionary learning As mentioned above, the spectrums in a single HS cube are similar to each other, and tailored dictionary is expected to perform well in this case. [29] propose a method of training sampling matrix and dictionary for a specific class of images. If, however prior knowledge on the data is not assumed, we may still be able to construct useful dictionary based on the sampled data itself. We call this approach Bootstrap method for dictionary construction. We plan to use PCA, ICA [27] and KSVD [25] for the dictionary construction. Preliminary experiments show promising results.

4) Spectral axis regularization As already hinted in subsubsection 3.3.6, we plan to explore methods for the regularizing the spectral axis that preserve spikes. We will test two variations on the total-variation

Figure 35: Comparison between the different reconstruction schemes - $\ell_1$, $TV_{xy}$-regularized $\ell_1$, and $\ell_1+TV_{xy}+I_{RGB}$ regularization. PSNR2 is compared between the different algorithms.
algorithm, one that uses the $\ell_p$ norm with $p < 1$, and the other will be an iteratively re-weighted total variation.

Acknowledgment: This research was supported in part by the European Community’s FP7-FET program, SMALL project, under grant agreement no. 225913.

References


[27] A. Bell, T. Sejnowski, The independent components of natural scenes are edge filters, Vision research, 37(23):Pages 3327-3338, 1997

58


[34] $\ell_1$ magic in E.Candes’s website: http://www.acm.caltech.edu/l1magic/


